UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Brain Inspired Neural Network Models of Visual Motion Perception and Tracking in Dynamic Scenes

Permalink

https://escholarship.org/uc/item/6f17b5s5

Author

Kashyap, Hirak Jyoti

Publication Date

2020

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Brain Inspired Neural Network Models of Visual Motion Perception and Tracking in Dynamic Scenes

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Hirak J. Kashyap

Dissertation Committee:
Professor Jeffrey L. Krichmar, Chair
Professor Nikil D. Dutt
Professor Charless C. Fowlkes
Professor Emre Neftci

2020

Chapter 5 © 2018 IEEE
Chapter 6 © 2018 IEEE
All other materials © 2020 Hirak J. Kashyap

DEDICATION

To maa and deuta

TABLE OF CONTENTS

Page

LIST OF FIGURES **vi**

LIST OF TABLES **xii**

ACKNOWLEDGMENTS **xiii**

VITA **xiv**

ABSTRACT OF THE DISSERTATION **xvii**

1 Introduction **1**

2 Background **5**
2.1 Visual Motion Processing in the Primate Brain 5
2.1.1 Two-stream Hypothesis 5
2.1.2 The Dorsal Visual Pathway 7
2.1.3 Neural Correlates of Predictive Smooth Pursuit Eye Movement 9
2.2 Models of the Visual System 10
2.2.1 Hierarchical Models 10
2.2.2 Computational Models of Dorsal Visual Pathway 11
2.2.3 Convolutional Neural Networks 13
2.3 Algorithms for Motion Estimation 16
2.3.1 Ego-motion Estimation Methods 16
2.3.2 Object Motion Estimation Methods 17
2.3.3 Motion Field Model 17
2.3.4 Flow Parsing 20

3 Sparse Representations for Object and Ego-motion Estimation in Dynamic Scenes **22**
3.1 Introduction 22
3.2 Methods 25
3.2.1 Representation of Ego-motion Using a Sparse Basis Set 25
3.2.2 Joint Optimization for Basis Vectors and Coefficients 30
3.3 Experimental results 35
3.3.1 Datasets 36

3.3.2 Training . . . 37
3.3.3 Ego-motion Prediction . . . 37
3.3.4 Object-motion Prediction . . . 40
3.3.5 Sparsity Analysis . . . 43
3.3.6 The Learned Basis Set . . . 46
3.4 Discussion . . . 47

4 Convolutional Neural Network Model of Cortical Visual Motion Perception 50
4.1 Introduction . . . 50
4.2 Methods . . . 51
4.2.1 Convolutional Neural Network Model . . . 51
4.2.2 Visual Stimuli . . . 55
4.2.3 Training . . . 56
4.2.4 Neuron Activation Analysis . . . 57
4.3 Results . . . 59
4.3.1 Neuronal Response Additivity . . . 59
4.3.2 Population Response Interactions with Combined Stimuli . . . 63
4.3.3 Stimulus Specific Selectivity . . . 67
4.4 Discussion . . . 73

5 Recurrent Neural Network Model of Pursuit Eye Movement for Visual Tracking 77
5.1 Introduction . . . 77
5.2 Methods . . . 79
5.2.1 Neuron Model and Network Architecture . . . 80
5.2.2 Online Learning . . . 83
5.3 Experimental Results . . . 84
5.3.1 Pursuit Initiation . . . 85
5.3.2 Predictive Pursuit . . . 87
5.3.3 Unpredictable Perturbation and Phase Shift . . . 90
5.3.4 Unpredictable Target Velocity . . . 91
5.4 Discussion . . . 92
5.4.1 Other Computational Models of Smooth Pursuit . . . 93
5.4.2 The role of FEF in Predictive Pursuit . . . 95

6 A Fully Neuromorphic Stereo Vision System for Dynamic scenes 96
6.1 Introduction . . . 96
6.2 Neuromorphic Hardware . . . 100
6.2.1 Dynamic Vision Sensors . . . 100
6.2.2 TrueNorth Processor . . . 101
6.3 Methods . . . 103
6.3.1 Rectification . . . 104
6.3.2 Multiscale Temporal Representation . . . 104
6.3.3 Morphological Erosion and Dilation . . . 106

6.3.4 Multiscale Spatiotemporal Features 106
6.3.5 Hadamard Product 107
6.3.6 Winner-take-all 107
6.3.7 Consistency Constraints 110
6.4 Experimental Results 110
6.4.1 Datasets 110
6.4.2 Results 113
6.5 Discussion 114

7 Conclusion **117**
7.1 Summary 117
7.2 Future Directions 118

Bibliography **120**

LIST OF FIGURES

Page

1.1 Anatomy of the visual motion pathway in the macaque brain (reprinted with permission of Annual Reviews, Inc. from (Britten, 2008)). **a**, Simplified schematic of the connections between areas known to play a part in motion analysis. **b**, Anatomical locations of the areas on a slightly "inflated" monkey brain to allow visualization of areas within sulci. The viewpoint is dorsal and lateral. Nomenclature and area boundaries after Felleman and Van Essen (1991); image generated with public domain software CARET (http://brainmap.wustl.edu/caret; (Van Essen et al., 2001)). 2

2.1 The two cortical visual processing pathways in the primate brain. 6
2.2 Different stages of motion processing (adapted from Nishida et al. (2018)). Representative image and optic flow from the flower garden sequence (Wang and Adelson, 1994). 12
2.3 Hierarchical organization of simple and complex cells in the visual system as found by Hubel and Wiesel (1962) (a) and the convolution and pooling operations in a CNN (b). Diagram is adapted from Lindsay (2020). The dashed circles in (a) denote the receptive fields of simple cells. 14

3.1 L0, L1, and L2 norm penalties and the proposed sharp sigmoid penalty for basis coefficient α_j. It can be observed that for $\alpha_j \geq 0$, the sharp sigmoid penalty approximates the L0 penalty and is continuous and differentiable. The sharp sigmoid function shown above corresponds to $Q = 25$ and $B = 30$. The L1 and L2 norm penalties enforce shrinkage on larger values of α_j. Moreover, for a set of coefficients, L1 and L2 norm penalties cannot indicate the number of $\alpha_j > 0$ due to not having any upper bound. 27
3.2 Derivative of the sharp sigmoid penalty function $p(\alpha_j)$ with respect to coefficient α_j. 31
3.3 Architecture of the proposed SparseMFE network. Conv blocks are fully convolutional layers of 2D convolution and ReLU operations. The receptive field size is gradually increased such that each neuron in the Conv1X-4 layer operates across the entire image. Outputs of all Conv blocks are non-negative due to ReLU operations. K, S, and P denote the kernel sizes, strides, and padding along vertical and horizontal directions of feature maps. F denotes the number of filters in each layer. The weights of the fully connected layer forms the basis for translational and rotational egomotion. 33

3.4 Qualitative results of SparseMFE on Sintel test split. The red colored overlay denotes the dynamic region masks. 42
3.5 Qualitative results of SparseMFE on KITTI benchmark real world frames (Menze and Geiger, 2015). Ground truth OMF is not available, however, ground truth dynamic region masks are provided in the benchmark. The ground truth depth map is sparse, and the pixels where depth is not available are colored in black. 42
3.6 Neuron activation profile in the bottleneck layer on Sintel test split for different types of sparsity regularization. (a) Number of nonzero activations in the bottleneck layer for frame sequences in the Sintel test split. Line colors denote the sparsity regularization used. (b) Activation heatmap of the bottleneck for the *market_5* frame shown in Figure 3.4. All experiments are conducted after the network has converged to a stable solution. 43
3.7 Qualitative OMF and dynamic mask prediction results comparing L1, L2, and Sharp Sigmoid sparsity penalties, in terms of their robustness to removal of bottleneck layer neurons during testing. 44
3.8 Ablation experiment to study the effect of the sparsity loss coefficient λ_s on ego-motion prediction. During test, only a fraction of the bottleneck layer neurons are used for ego-motion prediction based on activation magnitude and the rest are set to zero. ATE is averaged over all frames in KITTI test sequences 09 and 10. 45
3.9 Projection of the learned EMF basis set for rotational and translational ego-motion to the Euclidean space in the camera reference frame. The dots represent the learned bases and the solid lines represent the positive X, Y, and Z axes of the Euclidean space. The red circles indicate a pair of translation and rotation bases that share a same coefficient. 46

4.1 The convolutional neural network used to simulate the MSTd-like model neurons that respond to interactions between object motion and optic flow. The network receives 2D optic flow and pixel-wise depth map as input and predicts 6DoF ego-motion parameters, pixel-wise 2D object motion, and dynamic object mask. The example in the figure shows a stimulus made of counter-clockwise observer translation along the horizontal plane with speed $3.14m/s$, observer rotation $1°/s$ w.r.to the vertical axis, and simultaneous clock-wise object motion ($3.14m/s$). The convolutional layers output a feature activation matrix F, which is used to decode the three outputs. 52

4.2 Configuration of the virtual scene and the motion patterns used to generate visual stimuli (adapted from (Sato et al., 2010)). (a) Optic flow and object motion visual stimuli simulate translational movement of the observer along the red circular trajectory of 8 meter diameter, in front of earth fixed cloud of dots. The two-part object consists of a triangle and circle and moves along the blue circular trajectory to simulate self-motion. (b) The stimulus conditions containing either object motion or optic flow in clockwise or anticlockwise direction. (c) Sixteen stimulus conditions combining four clockwise/anticlockwise direction combinations with four phase rotations between optic flow and object motion. The locations of the arrows denote the initial positions of the observer and object derived from the phase difference between optic flow and object motion. 54
4.3 The loss terms averaged over all training samples during different epochs of training. Loss values are represented in arbitrary units. 56
4.4 Distribution of normalized response additivities of the population of MSTd-like model neurons for the sixteen combined stimulus conditions (a, rows 1 - 4) and the four relative directions accumulated across the four relative phase conditions (a, row 5) and of MSTd neurons (Reprinted from Sato et al. (2010)) for the four relative directions accumulated across the four relative phase conditions (b). In each subplot, response additivity (abscissa) is plotted as the difference between normalized combined and summed responses of all neurons across all response intervals (ordinate) in that stimulus condition. . 60
4.5 Distributions of normalized response additivity of MSTd and model neurons across all sixteen combined stimulus conditions and the corresponding shifted distributions after baseline subtraction. The green bars represent distributions before subtracting baseline neuron activations and the red/blue bars represent distributions after subtracting baseline activations. MSTd response additivity distributions were generated using the mean and standard deviation provided by Sato et al. (2010) across all sixteen combined stimulus conditions. 62
4.6 The distribution of regression fits of responses to combined stimuli in terms of responses to optic flow and object motion stimuli, their multiplicative interactions, and other stimulus parameters. (a) The distribution of fits produced by MSTd neuron responses (reprinted from (Sato et al., 2010)). Other frames depict the distribution fits based on model neuron responses (b), after average baseline subtraction (c), after blank baseline subtraction (d), with multiplicative factors (e), and with additional stimulus parameters (f). 64
4.7 Population response additivity of MSTd neurons (a, reprinted from (Sato et al., 2010) and of MSTd-like model neurons before (b) and after (c, d) baseline subtraction. In each panel, the blue dashed line denotes the additivity line, the region above it represents super-additivity and the region below it represents sub-additivity. The solid red line is the regression fit for combined responses in terms of the summer object-only and flow-only responses. Responses are normalized to the largest response elicited by the neuron for any of the object-only, flow-only, or combined stimuli. 66

4.8 MSTd-like neuron that responds to combined object and self-motion stimuli. Circular activation plots of responses by a MSTd-like model neuron to 20 combined and alone stimulus conditions. Location around each circle corresponds to the position of the observer on a circular trajectory of self-movement when the activation was recorded. All activations are normalized to the maximum activation of the neuron across all stimulus conditions. (a) activations for object motion and optic flow stimuli, (b) activations for object motion and optic flow stimuli after blank baseline subtraction, (c) activations for sixteen combined stimulus conditions, (d) activations for sixteen combined stimulus conditions after blank baseline subtraction. 68
4.9 Circular activation plots of activations of four MSTd-like model neurons. Each neuron prefers either object motion (a), ego-motion (b), or both ego-motion and combined stimuli (c), or has no preference (d). Activations are shown after blank baseline subtraction. 69
4.10 Stimulus specific selectivity of all neurons to 20 combined and alone stimulus conditions after blank baseline subtraction. Location around each circle corresponds to the position of the observer on a circular trajectory of self-movement simulated using visual stimuli. Each filled circle in each plot represents one neuron, its angular location defines the preferred motion stimulus corresponding to that location, its radial distance defines the normalized activation magnitude to the preferred stimuli. Area of each circle is proportional to its radial distance. Each neuron is color tagged from the colorbar shown below and is consistent across all circular plots. (a) Selectivity for object motion and optic flow stimulus conditions. (b) Selectivity for the sixteen combined stimulus conditions. 71

5.1 The proposed model for predictive smooth pursuit eye movement generation in primates. The plausible brain regions performing the specific functions in the pursuit pathway are shown in green colored boxes. The retinotopic RS is extracted from visual field with a time delay of δ by the dorsal visual pathway (RS^{δ}). A recurrent network of neurons (blue circles) in the FEF region uses RS^{δ} to learn the target velocity sequence and generates $\tilde{u}$, which is then low pass filtered by a leaky integrator to obtain eye velocity predictions ($\tilde{v}_E$). All red colored synaptic connections are modified during learning. Cerebellum and Brainstem together implement an inverse dynamic controller to generate the final eye velocity (v_E) via occulomotor control. 80
5.2 Eye velocity during pursuit initiation in response to a ramp stimulus of constant velocity 20 deg/s. The black dashed line depicts the target velocity. The colored lines are the eye velocity responses generated by the proposed model in 20 trials. 85

5.3 Mean eye acceleration versus target velocity between 80 ms and 180 ms after pursuit onset. Blue circles correspond to predictions by the proposed model and red circles correspond to experimental data by de Brouwer et al. (2002), reproduced from (de Xivry et al., 2013). Vertical bars are the standard deviations from mean. Experimental data is not available for target velocity -50 deg/s. 87
5.4 The pursuit eye velocity generated by our model in response to sinusoidal target velocity pattern. The black dashed line is the target velocity and the colored lines are the eye velocity simulated using the proposed model in different trials. The grey areas are the time periods where the target is occluded. (a) The target is always visible, (b) the target is temporarily occluded and then reappears, and (c) the target is permanently occluded after 15 seconds. 88
5.5 Response of the predictive pursuit model to unpredictable perturbation and phase shift. Black dashed line is the target velocity and the colored lines are the eye velocity generated by the model in 5 trials. R = 0.58 s is the experimental reaction time since perturbation calculated using the formula provided by Van den Berg (1988). Compares to Figure 8(a) of Van den Berg (1988). Pursuit starts at 0 s (not shown). 90
5.6 Eye velocity prediction by our model in response to an unpredictable target velocity input. The black dashed line is the target velocity sequence and the colored lines are the model output during ten trials. The grayed regions are occlusions. 91
5.7 Mean RS (the solid black trace) from 10 trials of the experiment shown in Figure 5.4a. The target velocity (the dashed line) is superimposed for reference. The RS signal is received by the predictive model after 80 ms to simulate sensory delays. 93

6.1 Frame based (a) and event-based (b-left DAVIS and c-right DAVIS) camera output for a rotating fan. Green dots are ON events, i.e. an increase in pixel intensity, and red dots are OFF events. 100
6.2 The time-stamp synchronized stereo rig is connected to a cluster of TrueNorth chips via ethernet. 101
6.3 The pipeline of execution using input events generated by left and right sensors. A toy example of main operations performed is demonstrated side-by-side in a single spatiotemporal scale, with a event on the left image and its two candidate corresponding events on the right image. Standard morphological operations and left-to-right consistency check are not demonstrated. 103
6.4 The WTA circuit and an example of operation. The bias -1 and control signals in Equation 6.9 are not shown. 109

6.5 Experimental results obtained using TrueNorth. a) Example synthetic depth pattern, b) ground truth depth map, c) random dot stimuli (RDS), d) depth map superimposed on RDS, e) depth map obtained from corelet implementation, f) corelet result after erosion and dilation, g) fan sequence input received from the left-right DAVIS cameras and results generated by each layer of corelets from this input, h) example frame with fan rotating in a particular orientation, i) 3D reconstruction by the proposed method as seen from an angled front view (screen capture from the 3D visualizer), j) 3D reconstruction from a top view, k-l) Kinect depth maps with static fan blades, m) merged Kinect depth map, n-p) butterfly rotating around the spring base, q-r) Kinect depth map for the butterfly frames, s) merged Kinect depth maps, t) left DAVIS output for a 3 ms time window, u) right DAVIS output during the same window, v-y) top view from the 3D visualizer of 3D reconstruction four consecutive frames in the sequence at the butterfly rotates clockwise 111
6.6 Depth reconstruction of the fan (first column) and butterfly sequence (second column), each shown from two viewpoints. Each point in the butterfly sequence shown is the median coordinate estimate of the butterfly location at a distinct time instant. 113

LIST OF TABLES

Page

3.1 Absolute Trajectory Error (ATE) on the KITTI visual odometry test set . . 38
3.2 Relative Pose Error (RPE) comparison on the Sintel test set 39
3.3 End Point Error (EPE) comparison of OMF prediction on the Sintel test split 41

4.1 Categorization of MSTd-like model neurons as having preference to one of the three stimulus categories: object motion, optic flow, and combined stimuli. . 72

6.1 Comparison of event-based depth estimation literature 115

ACKNOWLEDGMENTS

Foremost, I would like to thank my advisor Professor Jeff Krichmar for his guidance, support, inspiration, and endless patience over the last five years. I will always remain indebted to him for facilitating a high spirited PhD learning experience. I sincerely thank Professor Charless Fowlkes, Professor Nikil Dutt, and Professor Emre Neftci for mentoring me and sharing their expertise, for agreeing to join my dissertation committee, and for constructive feedback on my dissertation work. I would also like to thank Professor George Sperling for serving on my PhD candidacy committee and challenging me to approach my ideas from many perspectives.

I also owe my thanks to the following people and/or organizations.

The Cognitive Anteater Robotics Laboratory (CARL). Tiffany Hwu, Xinyun Zou, Jinwei Xing, Kexin Chen, Georgios Detorakis, Michael Beyeler, Ting-Shuo Chou, John Szura, Thea Weiss, and Stas Listopad for sharing their interesting viewpoints and collaborating on diverse projects. Meropi Topalidou, Emily Rounds, John Shepanski, Veronica Newhart, Philippe Gaussier, and Will Browne for advice on research and navigating academia. I enjoyed working with the CARL undergraduates, Giselle Tian, Lara Mirzakhanian, Steven Seader, Mylen Cruzado, Grace Pan, and Jueying Li, and thank them for assisting in our research projects.

Members of the Computational Vision group, Dutt Research Group, and Neuromorphic Machine Intelligence group for sharing their expert viewpoints and thoughtful discussions.

IBM Research - Brain Inspired Computing group for supporting the neuromorphic stereo vision project over two summer internships. Alex Andreopoulos and Myron Flickner for their supervision of the research presented in Chapter 6.

The funding sources that supported Ph.D. studies: the National Science Foundation (NSF), Toyota Motor North America, the Defense Advanced Research Projects Agency (DARPA), and Intel.

SciDraw.io for providing the brain outline.

My family for supporting my dreams and for being always there with me physically or remotely. Trina Krichmar and Jugal Kalita for care and encouragement.

My friends Aniket, Pingu, Jugal, Arkaditya, Niranjan, Shreyansh, and Tiago for going out of their ways to help me and making these years fun.

Finally, I would like to thank my partner and best friend Tulika for keeping me on track, pulling me out of many saddle points, and for sharing the journey with love and care.

VITA

Hirak J. Kashyap

EDUCATION

Doctor of Philosophy in Computer Science **2020**
University of California, Irvine *Irvine, California*

Master of Technology in Computer Science & Engineering **2014**
National Institute of Technology, Rourkela *Rourkela, India*

and Instituto Superior Técnico *Lisbon, Portugal*

Bachelor of Technology in Computer Science & Engineering **2012**
Tezpur University *Tezpur, India*

WORK AND RESEARCH EXPERIENCE

Graduate Student Researcher **2015–2020**
University of California, Irvine *Irvine, California*

Research Intern **2017**
IBM Research - Brain Inspired Computing *San Jose, California*

Research Intern **2016**
IBM Research - Brain Inspired Computing *San Jose, California*

Senior Research Fellow **2014-2015**
Tezpur University *Tezpur, India*

Visiting Student Researcher **2013-2014**
INESC-ID *Lisbon, Portugal*

Summer Research Intern **2011**
Tata Institute of Fundamental Research *Mumbai, India*

TEACHING EXPERIENCE

Teaching Assistant **F'15, W'16, W'17, S'18, W'18, W'19, W'20**
University of California, Irvine *Irvine, California*

PUBLICATIONS

Kashyap, H. J., Fowlkes, C., & Krichmar, J. L. (In review). Sparse Representations for Object and Ego-motion Estimation in Dynamic Scenes.

Newhart, V., Kashyap, H. J., Hwu, T., Eccles, J., & Krichmar, J. L. (In review). Evaluation of Pan/Tilt Head, Autonomous Navigation, and Arm/Hand Hardware for Learning Environments.

Hwu, T., Kashyap, H. J., & Krichmar, J. L. (2020). Applying a Neurobiological Model of Schemas and Memory Consolidation to Contextual Awareness in Robotics. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN).

Balaji, A., Adiraju, P., Kashyap, H. J., Das, A., Krichmar, J. L., Dutt, N., & Catthoor, F. (2020). PyCARL: A Common PyNN Interface for GPU-Accelerated Biologically Plausible Spiking Neural Network Simulation. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN).

Andreopoulos, A.*, Kashyap, H. J.*, Nayak, T. K., Amir, A., & Flickner, M. D. (2018). A low power, high throughput, fully event-based stereo system. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 7532-7542). (* equal contribution)

Kashyap, H. J., Detorakis, G., Dutt, N., Krichmar, J. L., & Neftci, E. (2018). A Recurrent Neural Network Based Model of Predictive Smooth Pursuit Eye Movement in Primates. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN) (pp. 5353-5360).

Chou, T. S.*, Kashyap, H. J.*, Xing, J., Listopad, S., Rounds, E. L., Beyeler, M., Dutt, N., & Krichmar, J. L. (2018). CARLsim 4: an open source library for large scale, biologically detailed spiking neural network simulation using heterogeneous clusters. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN) (pp. 1158-1165). (* equal contribution)

Hoque, N., Kashyap, H., & Bhattacharyya, D. K. (2017). Real-time DDoS attack detection using FPGA. Computer Communications, 110, 48-58.

Kashyap, H., Ahmed, H. A., Hoque, N., Roy, S., & Bhattacharyya, D. K. (2016). Big data analytics in bioinformatics: architectures, techniques, tools and issues. Network Modeling Analysis in Health Informatics and Bioinformatics, 5(1), 28.

Kakati, T., Kashyap, H., & Bhattacharyya, D. K. (2016). THD-module extractor: an application for CEN module extraction and interesting gene identification for Alzheimer's disease. Scientific Reports, 6, 38046.

Kashyap, H. & Chaves, R. (2016). Compact and on-the-fly secure dynamic reconfiguration for volatile FPGAs. ACM Transactions on Reconfigurable Technology and Systems (TRETS), 9(2), 11.

Kashyap, H. & Chaves, R. (2014). Secure partial dynamic reconfiguration with unsecured external memory. In Proceedings of the 24th International Conference on Field Programmable

Logic and Applications (FPL).

Bhuyan, M. H., Kashyap, H. J., Bhattacharyya, D. K., & Kalita, J. K. (2013). Detecting distributed denial of service attacks: methods, tools and future directions. The Computer Journal, 57(4), 537-556.

Kashyap, H. J. & Bhattacharyya, D. K. (2012). A DDoS attack detection mechanism based on protocol specific traffic features. In Proceedings of the International Conference on Computational Science, Engineering and Information Technology (pp. 194-200)

ABSTRACT OF THE DISSERTATION

Brain Inspired Neural Network Models of Visual Motion Perception and Tracking in Dynamic Scenes

By

Hirak J. Kashyap

Doctor of Philosophy in Computer Science

University of California, Irvine, 2020

Professor Jeffrey L. Krichmar, Chair

For self-driving vehicles, aerial drones, and autonomous robots to be successfully deployed in the real-world, they must be able to navigate complex environments and track objects. While Artificial Intelligence and Machine Vision have made significant progress in dynamic scene understanding, they are not yet as robust and computationally efficient as humans or other primates in these tasks. For example, the current state-of-the-art visual tracking methods become inaccurate when applied to random test videos. We suggest that ideas from cortical visual processing can inspire real world solutions for motion perception and tracking that are robust and efficient. In this context, the following contributions are made in this dissertation. First, a method for estimating 6DoF ego-motion and pixel-wise object motion is introduced, based on a learned overcomplete motion field basis set. The method uses motion field constraints for training and a novel differentiable sparsity regularizer to achieve state-of-the-art ego and object-motion performances on benchmark datasets. Second, a Convolutional Neural Network (CNN) that learns hidden neural representations analogous to the response characteristics of dorsal Medial Superior Temporal area (MSTd) neurons for optic flow and object motion is presented. The findings suggest that goal driven training of CNNs might automatically result in the MSTd-like response properties of model neurons. Third, a recurrent neural network model of predictive smooth pursuit eye movements is pre-

sented that generates similar pursuit initiation and predictive pursuit behaviors as observed in humans. The model provides the computational mechanisms of formation and rapid update of an internal model of target velocity, commonly attributed to zero lag tracking and smooth pursuit of occluded objects. Finally, a spike based stereo depth algorithm is presented that reconstructs dynamic visual scenes at 400 frames-per-second with one watt of power consumption when implemented using the IBM TrueNorth processor. Taken together, the presented models and implementations provide the computations for motion perception in the dorsal visual pathway in the brain and inform ideas for efficient computational vision systems.

Chapter 1

Introduction

The dorsal visual pathway in the primate brain, which originates at the primary visual cortex (V1) and runs through the dorsal surface into the parietal cortex, is known to perform motion perception and object localization (Orban, 2008). As such, the computations performed by this pathway are highly important for goal-directed behavior in visual-spatial tasks, such as reaching and grasping (Mishkin and Ungerleider, 1982). Parts of the dorsal pathway were also found to be involved in generation of eye movements, such as, saccadic eye movement to fixate on a target and smooth pursuit eye movement to track a moving target (Ilg and Schumann, 2007). It should be noted that the visual capabilities of primates are not produced by the dorsal visual pathway alone. There is another ventral visual pathway that enables recognition and semantic categorization of objects and shapes (Gross et al., 1993). We have a relatively better understanding of neural encoding along the ventral pathway and many successful computational models relating to the regions in this pathway have been proposed (Fukushima, 1988; Riesenhuber and Poggio, 1999).

Given that primates are highly interactive natural agents relying on high acuity vision, the quest for developing equally efficient mobile robotic systems can greatly benefit from under-

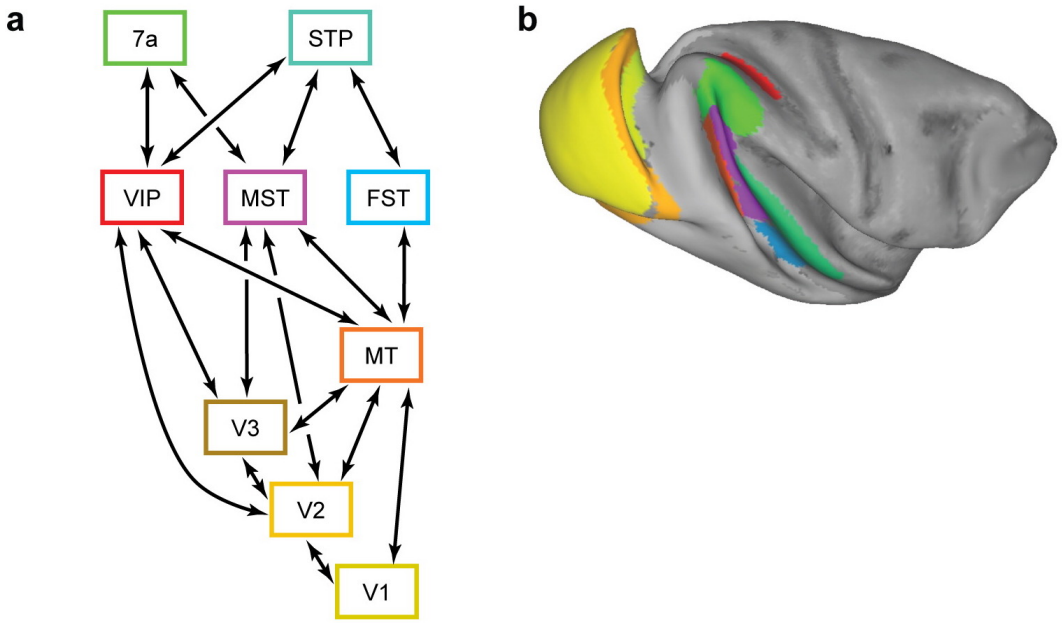


Figure 1.1: Anatomy of the visual motion pathway in the macaque brain (reprinted with permission of Annual Reviews, Inc. from (Britten, 2008)). **a**, Simplified schematic of the connections between areas known to play a part in motion analysis. **b**, Anatomical locations of the areas on a slightly "inflated" monkey brain to allow visualization of areas within sulci. The viewpoint is dorsal and lateral. Nomenclature and area boundaries after Felleman and Van Essen (1991); image generated with public domain software CARET (http://brainmap.wustl.edu/caret; (Van Essen et al., 2001)).

standing of the computations performed by the dorsal visual pathway. Navigational tasks require estimating the motion parameters of dynamic scene elements, which are computed by the higher order areas in this pathway. Similarly, augmented and virtual reality systems can benefit from incorporating the mechanisms of perceptual stability implemented by the dorsal visual pathway to create stable viewing conditions during eye, head, or body movements. This pathway is also involved in motion processing for visual tracking, enabling efficient tracking performance in cluttered environments. The existing visual tracking systems can be made robust by mimicking the predictive behaviors observed in primates.

Figure 1.1 depicts the prominent motion processing regions in the dorsal visual pathway and their interconnections. The neurons in the successive regions along the dorsal pathway respond to increasingly complex spatiotemporal stimuli. For example, a subset of neurons

in V1 respond to local motion of features, such as, oriented edges at multiple temporal and spatial frequencies (Orban et al., 1986). In the Middle Temporal area (MT/V5) at a later stage of the pathway, neurons are tuned to three dimensional speed gradients, possibly combining depth and motion (Xiao et al., 1997). MT/V5 have strong projections to the Medial Superior Temporal area (MST), which have neurons that respond to multiple patterns of large-field apparent motion on the retina or optic flow generated by observer self-rotation and translation and object movement in 3D (Duffy and Wurtz, 1991a,b; Graziano et al., 1994).

A major goal of this dissertation is to investigate how these neurons compute their responses to visual stimuli. From a mechanistic standpoint, we would like to know the series of computations performed by the dorsal pathway that can be implemented on a computer. For example, we know that the V1 simple cell responses can be expressed as Gabor filters (Adelson and Bergen, 1985) and the population level code is sparse and overcomplete (Olshausen and Field, 1997; Barlow, 1981). The MT neurons can be partially modeled as weighted summation over V1 responses with the same direction tuning (Simoncelli and Heeger, 1998; Rust et al., 2006). However, beyond that, computations performed by the higher order dorsal pathway areas are not fully understood. This could be due to complexity of the geometric problems they solve, which involve large and sometimes ambiguous combinations of stimulus parameters and experiments resort to a small subset of possible stimulus combinations. For example, separate studies have found the MST neurons to be selective to depth, self-translation, self-rotation, object motion direction and speed, however, a complete characterization of the interactions between responses to these components is missing (Sasaki et al., 2019, 2017; Takahashi et al., 2007; Duffy and Wurtz, 1991a,b; Graziano et al., 1994). As a result, the computational models engineered to match the MST neural responses are biased and cannot predict the coding principles of the neurons for naturalistic stimuli (Browning et al., 2009; Layton and Fajen, 2020; Grossberg et al., 1999).

In this dissertation, we use learning based parametric methods to explain neural responses for the higher order dorsal visual areas with carefully designed optimizations including biological constraints. These methods can be used to predict the underlying computational principles of cortical visual processing (Beyeler et al., 2016; Zemel and Sejnowski, 1995; Park et al., 2000). Recently, Convolutional Neural Networks (CNNs) with hierarchically organized receptive fields have been used to predict responses of deep layers of the ventral visual pathway (Güçlü and van Gerven, 2015; Yamins et al., 2014; Seeliger et al., 2018; Khaligh-Razavi and Kriegeskorte, 2014; Tripp, 2017). They may as well serve as a model of the dorsal visual pathway due to their scalability and hierarchical feature learning capabilities.

This dissertation presents neural network models of visual motion perception and tracking that combine ideas from cortical motion processing and artificial neural networks. Chapter 3 shows the use of motion field constraints and the number of nonzero neuron activations as a regularization term to learn an overcomplete ego-motion basis set using a CNN. Following up on this, Chapter 4 takes a more detailed approach of comparing CNN activations to the responses of individual neurons in the dorsal region of MST when presented with identical stimuli of object and ego-motion. Frontal Eye Fields (FEF) is a frontal cortical area that receives inputs from MST and is a major center for producing smooth pursuit eye movement signals for visual tracking of a moving object. Chapter 5 discusses how object velocity signal from MST, delayed by cortical processing, might be used by FEF neurons to generate eye velocity commands during occlusions of the target object. Chapter 6 demonstrates a spike based stereo algorithm for reconstruction of dynamic 3D scenes with fast moving objects. We summarize our findings and future directions of research in Chapter 7.

Chapter 2

Background

This chapter covers the neuroscience, modeling and methodological background for the remainder of the dissertation. It starts with a discussion of motion processing in the brain. Covers models of motion processing that are neurobiologically inspired, and those that are more computer science oriented. The chapter ends with a discussion of some of the mathematical techniques used by these models and by the model presented in the later chapters.

2.1 Visual Motion Processing in the Primate Brain

2.1.1 Two-stream Hypothesis

Two separate pathways of visual processing in the primate brain were first observed in monkeys with lesion experiments. It was observed that lesions of inferior temporal cortex caused severe deficits in visual recognition tasks, but did not affect performance on visuospatial tasks, such as, reaching and grasping. In contrast, parietal lesions did not affect visual recognition performance, however, caused severe deficits in visuospatial tasks. Based on

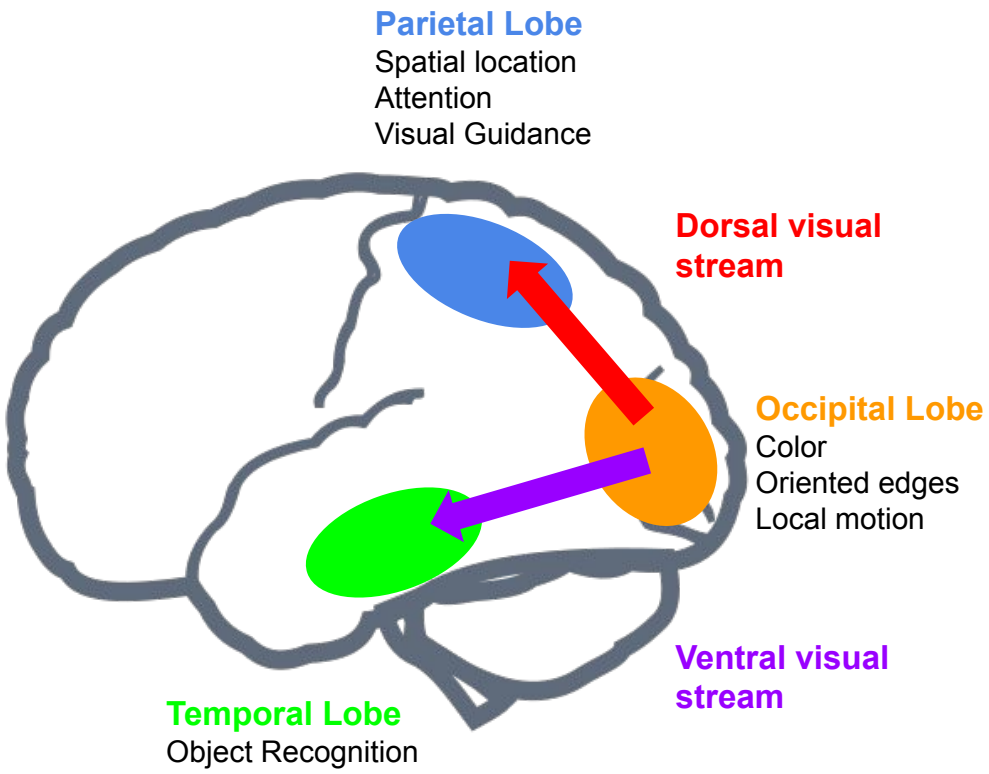


Figure 2.1: The two cortical visual processing pathways in the primate brain.

these findings, (Mishkin et al., 1983) proposed the existence two separate pathways in the visual cortex: an occipitotemporal "what" pathway or ventral stream that facilitates recognition of an object and an occipitoparietal "where" pathway or dorsal stream that facilitates perception of where objects are and visual guidance toward them. Comparable visual pathways were later found in human brain as well (Newcombe et al., 1987). The two pathways are shown in Figure 2.1.

Response properties of individual neurons and neuron populations are distinct between dorsal and ventral pathways. Neurons in the dorsal pathway areas, such as V1, V2, V3, V3A, MT/V5, MST, and additional areas in the inferior parietal cortex, respond to spatial aspects of the visual stimulus, such as direction and speed of the stimulus (Orban, 2008). On the other hand, neurons in the ventral pathway areas, such as V1, V2, V4, and inferior temporal areas TEO and TE, are selective visual features that identifies an object, such as color,

shape, texture, and object parts (Grill-Spector and Weiner, 2014).

2.1.2 The Dorsal Visual Pathway

We only discuss the dorsal visual pathway areas in detail, since this pathway is primarily involved with cortical motion processing, which is most pertinent to this dissertation.

V1

V1 contains neurons selective to direction and speed of visual stimulus. Hubel and Wiesel (1962) first discovered direction selective cells in V1. Speed tuning was discovered using moving bars (Orban et al., 1986). Using gratings stimuli, Priebe et al. (2006) was able to show the differences between speed tuning of V1 simple and V1 complex cells. The speed tuning in V1 complex cells is somewhat similar to that of MT neurons, however it can differ for other stimuli types (Priebe et al., 2006).

V2

The direction selective neurons in V2 have antagonistic surround and respond to motion contrast (Lu et al., 2010; Hu et al., 2018). They have smaller receptive field than MT neurons and are capable of performing figure-ground segregation. Hu et al. (2018) employed single-cell recordings of V2 to find high concentration of these motion contrast sensitive cells.

V3 and V3A

V3A area is adjacent to V3 and considered to independent from V3 (Essen and Zeki, 1978; Felleman and Van Essen, 1991). Braddick et al. (2001) found V3/V3A neurons to be selective

to coherent motion patterns using human fMRI studies. V3 is considered to be part of early visual areas based on its strong responses to local bar stimulus, whereas V3A is considered to be part of higher order visual cortex since it responds strongly to global motion stimulus (Hughes et al., 2019).

MT/V5

Middle Temporal area (MT) or V5 is probably the most studied area for motion processing in the primate brain. It receives inputs from V1, V2, and V3 (Ungerleider and Desimone, 1986). Large portions of the cells in MT are tuned to speed and direction of moving stimuli (Maunsell and Van Essen, 1983). The functional differences between MT neurons and early visual area neurons selective to direction and speed of local stimuli were initially unclear. Later studies found that MT neurons are tuned to three dimensional speed gradients, possibly combining depth and motion, whereas V1 and V2 were not (Xiao et al., 1997; DeAngelis and Newsome, 1999).

MST

MT outputs are projected to the Medial Superior Temporal area (MST). Neurons in MST respond to multiple patterns of optic flow and retinal motion generated by moving objects (Duffy and Wurtz, 1991a,b; Graziano et al., 1994; Sato et al., 2010). Neurons in the dorsal subdivision of MST, called MSTd, have large receptive fields and respond to apparent motion on the retina due to observer translation and rotation (Duffy and Wurtz, 1991a). On the other hand, neurons in the ventral subdivision of MST, called MSTv, have smaller receptive field and proposed to be encoding object motion (Eifuku and Wurtz, 1998). However, how these two subdivisions separate object and ego-motion is not known and there may not be any functional difference at all, as MSTd neurons have been found to respond to object

motion as well (Sato et al., 2010; Sasaki et al., 2017).

2.1.3 Neural Correlates of Predictive Smooth Pursuit Eye Movement

In the frontal cortex, the Frontal Eye Fields (FEF) show neuronal responses related to predictive smooth pursuit eye movement (Xiao et al., 2007). Single unit recordings in FEF have found neurons involved in predicting future target trajectories (Fukushima et al., 2002). Chou and Lisberger (2004) concluded that these neurons learn the target trajectory for prediction and are not related to pursuit adaptation. FEF acts as the main output center for eye movements, combining reactive and predictive components (Keating, 1993). The Supplementary Eye Field (SEF) are another frontal area related to predictive eye movements and it is reciprocally connected to the FEF area. SEF neurons are found to regulate prediction activity in FEF, such as release of the predictive component, gain modulation, and response timing, rather than signaling pursuit output to the motor neurons directly (Shichinohe et al., 2009). In SEF, the same population of neurons encode both pursuit and saccadic responses and is a likely candidate for combined pursuit and saccade prediction (Nyffeler et al., 2008).

FEF receives object velocity signals from the MST area, which responds to object motion. Lesions of MST and the preceding MT area caused pursuit deficit (Dursteler and Wurtz, 1988). Also, efferents from FEF are transmitted to dorsal pontine nuclei (PN) and reticularis tegmenti pontis (NRTP) regions in the brainstem. These brainstem regions relay information from FEF to cerebellum for oculomotor adaptation (Takagi et al., 2000). Based on the outcomes of lesion studies and anatomical connections, Keating (1991) suggested that FEF follows the parietal areas of the visual pathway and prior to PN in controlling pursuit eye movements.

Although many sub-cortical areas have been weakly linked to smooth pursuit and saccades,

PN plays a strong role in both. Almost half of the neurons in dorsal PN indicate signal related to pursuit and saccades (Dicke et al., 2004). The major role of this area is to relay signals received from cerebral cortex to cerebellum, which plays a major role in oculomotor adaptation (Gaymard et al., 1993). Another region in the brainstem that is involved in relaying pursuit signal to cerebellum is the nucleus reticularis tegmenti pontis (NRTP). This area particularly relays information from FEF and SEF to the flocculus-paraflocculus complex and the posterior vermis, the two major smooth pursuit representation areas in cerebellum (Stanton et al., 1988; Glickstein et al., 1994). In cerebellum, the flocculus-paraflocculus complex is primarily associated with coordination of eye-head movements generated by pursuit with those generated by vestibular reflexes (Rambold et al., 2002). These cerebellar areas directly project to eye-head motorneurons in the vestibular nuclei (Roy and Cullen, 2003).

2.2 Models of the Visual System

2.2.1 Hierarchical Models

Hubel and Wiesel (1962) found a hierarchical receptive field organization between simple and complex cells of primary visual cortex (V1) of cats responding to specific orientations of a bar. The simple cells respond to bars of light at specific orientations and locations. The complex cells similarly had preferred orientations, however responded just as strongly to similar bars located at nearby spatial locations. They concluded that complex cells were likely receiving inputs from simple cells with a specific preferred orientation and nearby preferred locations. Fukushima (1988) transformed this finding into a computational model of the ventral pathway of visual cortex, called Neocognitron, by stacking layers of simple and complex cells alternatively. After several iterations of this structure, a hierarchical model of the ventral visual pathway was created. The network weights in this model self-organized

based on repeated exposure to unlabeled images.

Many of following computational models maintained the hierarchical architecture (Riesenhuber and Poggio, 2000). Of those, HMAX by Riesenhuber and Poggio (1999) is one of the most successful. The model differentiated by the use a max operation to pool over simple cell responses as inputs to the complex cells. The model was consistent with neurophysiological data and matched human observers during initial 100-150 ms on a rapid object recognition task (Serre et al., 2007).

While these models are well established and explain neurophysiology data of visual processing, the obvious question that arises is why hierarchy is needed. An evolutionary explanation is that hierarchical organization require minimal rewiring in response to changing environmental conditions (Meunier et al., 2010). An alternative explanation revolves around decomposition of a complex task into multiple stages of simpler visual recognition tasks invariant to specific stimuli (Serre, 2014). Hierarchical organization also leads to efficient use of computational resources, since each lower level feature detectors can be reused in multiple recognition tasks (Geman, 1999). This argument is strengthened by the facts that the hierarchical organization is not limited to only the visual pathways and the brain is energy efficient.

2.2.2 Computational Models of Dorsal Visual Pathway

Visual motion perception can be divided into two levels, lower level processing and higher level processing (Nishida et al., 2018). In the lower level processing, local feature motion or optic flow is calculated and in the higher level processing, representations of the dynamic scene are extracted at high level from optic flow, such as self-motion, object-motion, object shape, biological motion, mechanical object properties, among others. Figure 2.2 illustrates the two stages.

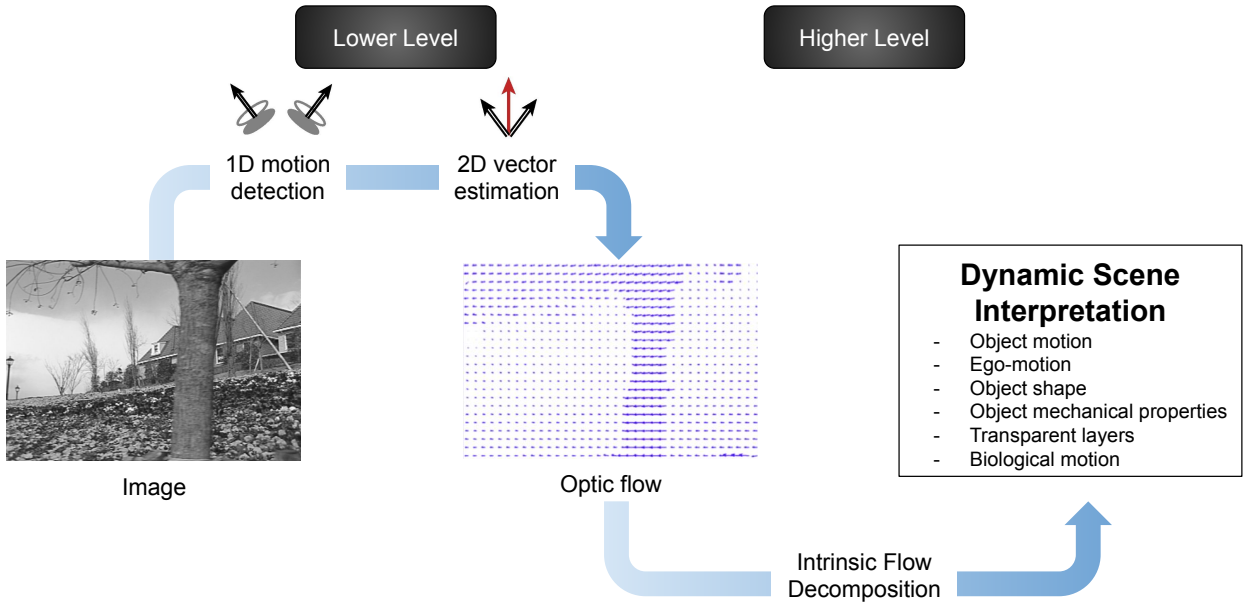


Figure 2.2: Different stages of motion processing (adapted from Nishida et al. (2018)). Representative image and optic flow from the flower garden sequence (Wang and Adelson, 1994).

In the lower level of motion processing, motion detectors detect motion in one and two dimensions, e.g. the Reichardt detector (Reichardt, 1957). A detector for one dimensional patterns can be thought of as a spatiotemporal orientation detector (Adelson and Bergen, 1985). The motion energy model derives motion signal from non-linear combinations of linear filter outputs (Adelson and Bergen, 1985). Based on this model, Simoncelli and Heeger (1998) proposed a model of area MT, where each MT neuron receives V1 motion energy signals from the plane with same speed and direction, and negative inputs from the planes further away.

Motion integration across multiple orientations gives rise to aperture problem. If the moving pattern is seen as a 1D contour inside an aperture, then image motion vector can be only determined along the component orthogonal to the contour. This problem can be solved by popuation coding of 1D motion signals across different orientations (Adelson and Movshon, 1982). Cross orientation integration of motion energies on a common velocity plane has been

observed experimentally for MT neurons (Nishimoto and Gallant, 2011; Rust et al., 2006). This suggests that the MT model by Simoncelli and Heeger (1998) is biologically plausible.

Although, the V1-MT feedforward model can explain some aspects of MT neural responses, it is not a sufficient model of human optic flow processing for complex and large magnitude optic flow (Hedges et al., 2011). On complex scenes, it has been recently shown that CNNs that learn to extract dense optic flow from image sequences procure representations somewhat similar to the cortical motion processors (Teney and Hebert, 2016; Nishida et al., 2018).

For the higher level motion processing stage, a template matching approach was first proposed by Tanaka and Saito (1989) to interpret the parameters of dynamic scenes. However, the model was too simple to account for the vast set of optic flows that can be generated by varying the parameters on the dynamic scene. If this problem were to be considered as a vector decomposition problem, then 2D optic flows need to be decomposed into 3D dynamic scene parameters. Currently, there is little information about how the dorsal visual areas achieve such decomposition. Grossberg et al. (2011) proposed a model of MST that uses directional grouping and competition to achieve such decomposition. However, the biological support for the model is unknown. Beyeler et al. (2016) suggested the decomposition results from a sparse signal representation mechanism with non-negativity constraints. Many other models hand design the MSTd response properties based on experimental data, which do not provide new insight about the computational mechanisms of optic flow decomposition into dynamic scene parameters (Browning et al., 2009; Layton and Fajen, 2020; Grossberg et al., 1999).

2.2.3 Convolutional Neural Networks

Much of the progresses made in the past decade in the field of computational vision are driven by deep convolutional neural networks (CNNs) (LeCun et al., 1989). These networks learn

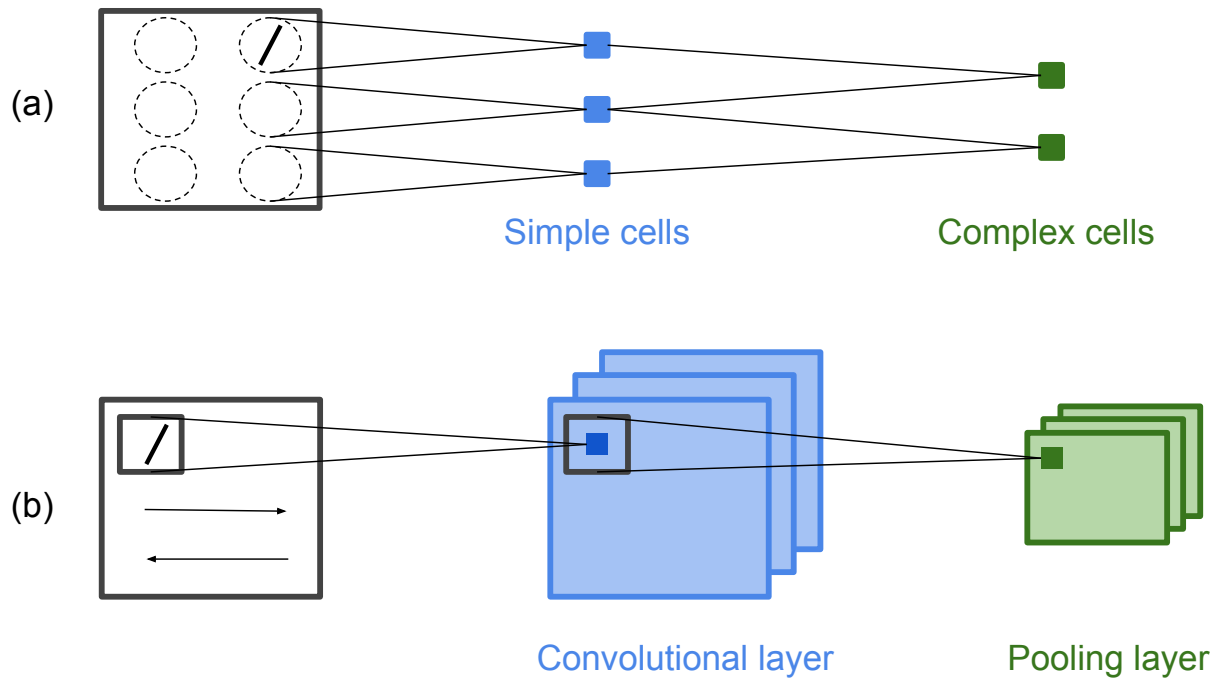


Figure 2.3: Hierarchical organization of simple and complex cells in the visual system as found by Hubel and Wiesel (1962) (a) and the convolution and pooling operations in a CNN (b). Diagram is adapted from Lindsay (2020). The dashed circles in (a) denote the receptive fields of simple cells.

useful features in visual input to solve many vision problems better than using handmade features. Examples include image classification (Krizhevsky et al., 2012; He et al., 2016), object detection (Girshick et al., 2014; Redmon et al., 2016), segmentation (He et al., 2017; Long et al., 2015), and image-to-image synthesis (Gatys et al., 2016; Johnson et al., 2016; Ledig et al., 2017; Zhu et al., 2017). Its feature learning capabilities arise from layers of neurons with hierarchically organized receptive fields, a feature that differentiates it from other neural networks and has similarities to the early computational models of visual cortex (Hubel and Wiesel, 1962; Fukushima, 1988). However, unlike its predecessors, CNN has evolved over time without relating much to the visual cortex and mainly for machine vision applications.

After CNN became mainstream and demonstrated its capabilities of learning complex fea-

tures (Zeiler and Fergus, 2014) and the lack of an alternative powerful computational model for diverse tasks and images led to investigations of CNN as a computational model of the visual system. One popular use of CNN in neuroscience studies has been to predict activities of real neurons based on activities of the model neurons at a certain deep layer. This has been done with high accuracy after showing the same image to both the CNN and an animal. Yamins et al. (2014) predicted V4 neuron responses by regressing model neurons activations and found that networks that perform better on object recognition tasks are also better predictors of neuron responses. This and other similar studies also found that activations in the later layers of the network are better predictor of V4 responses than the activations in the early layers (Güçlü and van Gerven, 2015; Seeliger et al., 2018). For population level comparison of CNN to visual cortex, representational similarity analysis (RSA) was proposed by Kriegeskorte et al. (2008), which first creates a matrix of how dissimilar the responses of each population are for every pair of images and followed by correlation analysis between matrices for different population. RSA provides a flexible comparison tool between neuron populations of any models, since similarity matrix can be constructed for any type of responses, including behavioral performance. Using RSA, Khaligh-Razavi and Kriegeskorte (2014) showed that Alexnet (Krizhevsky et al., 2012) matches many higher order visual areas better than previous models of the visual system.

Overall, CNNs have been successfully used to model the neural responses of the higher order areas of the ventral pathway. This was not possible with previous models, due to complexity of responses in these areas compared to V1 (Lindsay, 2020).

2.3 Algorithms for Motion Estimation

2.3.1 Ego-motion Estimation Methods

Ego-motion algorithms are categorized as direct methods (Concha and Civera, 2015; Engel et al., 2014) and feature-based methods (Mur-Artal et al., 2015; Strasdat et al., 2010; Jaegle et al., 2016; Nistér, 2004; Hartley, 1997). Direct methods minimize photometric image reconstruction error by estimating per pixel depth and camera motion, however they are slow and need good initialization. On the other hand, feature based methods use feature correspondences between two images to calculate camera motion. The feature based methods can be divided into two sub-categories, the first category of approaches uses a sparse discrete set of feature points and are called discrete approaches (Mur-Artal et al., 2015; Hartley, 1997; Nistér, 2004). These methods are fast, but are sensitive to independently moving objects. The second category uses optic flow induced by camera motion between the two frames to predict camera motion, also known as continuous approaches (Jaegle et al., 2016; Giachetti et al., 1998; Campbell et al., 2005; Lee and Fowlkes, 2019). This approach can take advantage of global flow pattern consistency to eliminate outliers, although it requires correct scene structure estimate (Black and Anandan, 1996).

Deep neural networks have been used to formulate direct ego-motion estimation as a prediction problem to achieve state-of-the-art results. Zhou et al. (2017) proposed deep neural networks that learned to predict depth and camera motion by training with a self supervised inverse warping loss between the source and the target frames. This self supervised deep learning approach has since been adopted by other methods to further improve ego-motion prediction accuracy (Mahjourian et al., 2018; Godard et al., 2019; Yin and Shi, 2018; Yang et al., 2018).

2.3.2 Object Motion Estimation Methods

Compared to monocular ego-motion estimation, fewer methods have been proposed for object-motion estimation from monocular videos. 3D motion field or scene flow was first defined in (Vedula et al., 1999) to describe motion of moving objects in the scene. Many approaches use depth as an additional input. Using Red, Green, Blue pixels plus depth (RGBD) input, scene flow was modelled as piecewise rigid flow superimposed with non-rigid residual from camera motion in (Quiroga et al., 2014). In another RGBD method, dynamic region segmentation was used to solve static regions as visual odometry and the dynamic regions as moving rigid patches (Jaimez et al., 2017). All of these methods assume a rigidity prior and fail with increasingly non-rigid dynamic scenes. To mitigate this, 2D scene flow or pixel-wise object-motion was estimated as non-rigid residual optic flow in the dynamic segments through supervised training of a deep neural network (Lv et al., 2018).

For RGB input, Vijayanarasimhan et al. (2017) proposed neural networks to jointly optimize for depth, ego-motion, and fixed number of objects using inverse warping loss. Due to the inherent ambiguity in the mixture of motion sources in optic flow, an expectation-maximization framework was proposed to train deep neural networks to jointly optimize for depth, ego-motion, and object-motion (Ranjan et al., 2019). These methods were evaluated only qualitatively on datasets with limited object movements.

2.3.3 Motion Field Model

We would like to discuss the formation of retinal motion in response to locomotion, structure, and other moving elements of the scene. They will guide our method design for the reverse problem, i.e. interpretation of these components from the limited information provided by optic flow due to 3D to 2D projection. Here we analyze the geometry of instantaneous static scene motion under perspective projection. These equations were derived previously

(Longuet-Higgins and Prazdny, 1980; Heeger and Jepson, 1992; Jaegle et al., 2016). We extend those to illustrate their use in deriving a simplified expression of instantaneous velocities of independently moving objects.

Let us denote the instantaneous camera translation velocity as $t = (t_x, t_y, t_z)^T \in R^3$ and the instantaneous camera rotation velocity as $\omega = (\omega_x, \omega_y, \omega_z)^T \in R^3$. Given scene depth $Z(p_i)$ and its inverse $\rho(p_i) = \frac{1}{Z(p_i)} \in R$ at an image location $p_i = (x_i, y_i)^T \in R^2$ of a calibrated camera image, the image velocity $v(p_i) = (v_i, u_i)^T \in R^2$ due to camera motion is given by,

$$v(p_i) = \rho(p_i)A(p_i)t + B(p_i)\omega \tag{2.1}$$

where,

$$A(p_i) = \begin{bmatrix} f & 0 & -x_i \\ 0 & f & -y_i \end{bmatrix}, \quad B(p_i) = \begin{bmatrix} -x_i y_i & f + x_i^2 & -y_i \\ -f - y_i^2 & x_i y_i & x_i \end{bmatrix}$$

If p_i is normalized by the focal length f, then it is possible to replace f with 1 in the expressions for $A(p_i)$ and $B(p_i)$.

If the image size is N pixels, then the full expression of instantaneous velocity at all the points due to camera motion, referred to as ego-motion field (EMF), can be expressed in a compressed form as,

$$v = \rho A t + B\omega \tag{2.2}$$

where, A, B, and ρ entails the expressions $A(p_i)$, $B(p_i)$, and $\rho(p_i)$ respectively for all the N points in the image as follows.

$$v = \begin{bmatrix} v(p_1) \\ v(p_2) \\ \vdots \\ v(p_N) \end{bmatrix} \in R^{2N\times 1}, \quad \rho At = \begin{bmatrix} \rho_1 A(p_1)t \\ \rho_2 A(p_2)t \\ \vdots \\ \rho_N A(p_N)t \end{bmatrix} \in R^{2N\times 1}, \quad B\omega = \begin{bmatrix} B(p_1)\omega \\ B(p_2)\omega \\ \vdots \\ B(p_N)\omega \end{bmatrix} \in R^{2N\times 1}$$

Note that the rotational component of EMF is independent of depth.

The monocular continuous ego-motion computation uses this formulation to estimate the unknown parameters t and ω given the point velocities v generated by camera motion (Jaegle et al., 2016; Heeger and Jepson, 1992). However, instantaneous image velocities obtained from the standard optic flow methods on real data are usually different from the EMF (Lee and Fowlkes, 2019). The presence of moving objects further deviates the optic flow away from the EMF. Let us call the input optic flow as $\hat{v}$, which is different from v. Therefore, monocular continuous methods on real data solve the following minimization objective to find t, ω, and ρ.

$$t^*, \omega^*, \rho^* = \underset{t,\omega,\rho}{\operatorname{argmin}} \|\rho At + B\omega - \hat{v}\|^2 \tag{2.3}$$

Following Zhang and Tomasi (1999) and Jaegle et al. (2016), without loss of generality, the

objective function can be first minimized for ρ as,

$$t^*, \omega^*, \rho^* = \underset{t,\omega}{\operatorname{argmin}} \underset{\rho}{\operatorname{argmin}} \|\rho At + B\omega - \hat{v}\|^2 \tag{2.4}$$

Therefore, the minimization for t^* and ω^* can be performed as,

$$t^*, \omega^* = \underset{t,\omega}{\operatorname{argmin}} \left\| A^{\perp} t^T (B\omega - \hat{v}) \right\|^2 \tag{2.5}$$

where $A^{\perp}t$ is orthogonal complement of At. This resulting expression does not depend on ρ and can be optimized directly to find optimal t^* and ω^*.

2.3.4 Flow Parsing

In dynamic scenes, the independently moving objects generate additional image velocities. Therefore, the resulting optic flow can be expressed as the sum of the flow components due to ego-motion ($\hat{v}_e$) and object-motion ($\hat{v}_o$). Following this, Equation 2.5 can be generalized as,

$$t^*, \omega^* = \underset{t,\omega}{\operatorname{argmin}} \left\| A^{\perp} t^T (B\omega - \hat{v}_e - \hat{v}_o) \right\|^2 \tag{2.6}$$

Since $\hat{v}_o$ is independent of t and ω, it can considered as non-gaussian additive noise and Equation 2.6 provides a robust formulation of Equation 2.5. After solving for t^* and ω^*,

image velocity due to object-motion across the entire image can be recovered as,

$$\tilde{v}_o = \hat{v} - \rho A t^* + B\omega^* \tag{2.7}$$

We will refer to $\tilde{v}_o$ as the predicted object-motion field (OMF). Equation 2.7 is equivalent to flow parsing, which is a mechanism proposed to be used by the human visual cortex to extract object velocity during self movement (Warren and Rushton, 2009).

Note that the expression is dependent on inverse depth (ρ). Although human observers are able to extract depth in the dynamic segments using stereo input and prior information about objects, the existing monocular structure prediction methods cannot reliably estimate depth in the dynamic segments without prior information about objects (Mahjourian et al., 2018; Godard et al., 2019; Yin and Shi, 2018; Yang et al., 2018).

Chapter 3

Sparse Representations for Object and Ego-motion Estimation in Dynamic Scenes

3.1 Introduction

Object and ego-motion estimation in videos of dynamic scenes are fundamental to autonomous navigation and tracking, and have found considerable attention in the recent years due to the surge in technological developments for self-driving vehicles. The task of 6DoF ego-motion prediction is to estimate the six parameters that describe the three-dimensional translation and rotation of the camera between two successive frames. Whereas, object-motion can be estimated either at an instance level where each object is assumed rigid (Byravan and Fox, 2017) or pixel-wise without any rigidity assumption, that is, parts of objects can move in different directions (Vijayanarasimhan et al., 2017; Lv et al., 2018). Pixel-wise object-motion estimation is more useful since many objects in the real world, such as people,

are not rigid (Houenou et al., 2013).

In order to compute object velocity, the camera or observer's ego-motion needs to be compensated (Bak et al., 2014). Likewise, the presence of large moving objects can affect the perception of ego-motion (Stein et al., 2000). Both ego and object-motion result in the movement of pixels between two successive video frames, which is known as optic flow and encapsulates multiple sources of variation. Scene depth, ego-motion, and velocity of independently moving objects determine pixel movements in videos. These motion sources of optic flow are ambiguous, particularly in the monocular case, and so the decomposition is not unique (Ranjan et al., 2019).

Several different approaches for ego-motion estimation have been proposed. Feature based methods compute ego-motion based on motion of rigid background features between successive frames (Hartley, 1997; Fredriksson et al., 2015; Mur-Artal et al., 2015; Strasdat et al., 2010; Jaegle et al., 2016; Nistér, 2004). Another well studied approach is to jointly estimate structure from motion (SfM) by minimizing warping error across the entire image (Concha and Civera, 2015; Engel et al., 2014; Newcombe et al., 2011; Furukawa et al., 2010). While the traditional SfM methods are effective in many cases, they rely on accurate feature correspondences, which are difficult to find in low texture regions, thin or complex structures, and occlusion regions. Some of the issues with traditional SfM approaches have been tackled using deep neural network predictors trained with inverse warping loss (Zhou et al., 2017; Yin and Shi, 2018; Mahjourian et al., 2018; Godard et al., 2019). These deep learning methods rely on finding the rigid background segments for ego-motion estimation. However, these methods do not separate pixel velocities into ego and object-motion components. All of the prior methods that solve for both ego-motion and pixel-wise object motion use depth as an additional input (Quiroga et al., 2014; Jaimez et al., 2017; Lv et al., 2018). Joint estimation of object and ego-motion from monocular RGB frames can be ambiguous (Ranjan et al., 2019). However, the estimation of ego and object-motion components from their composite

optic flow could be improved by using the geometric constraints of motion-field to regularize a deep neural network based predictor (Heeger and Jepson, 1992; Jaegle et al., 2016).

In this chapter, we present a novel approach for predicting 6DoF ego-motion and pixel-wise image velocity generated by non-rigid moving objects in videos, considering motion-field decomposition in terms of ego and object-motion sources in the dynamic image segments. Our approach first predicts the ego-motion field covering both rigid background and dynamic segments, from which object-motion and 6DoF ego-motion parameters can be derived in closed form. Compared to the existing approaches, our method does not assume a static scene (Hartley, 1997; Zhang and Tomasi, 2002; Fredriksson et al., 2015) and does not require dynamic segment mask (Zhou et al., 2017; Lee and Fowlkes, 2019; Vijayanarasimhan et al., 2017) or depth (Lv et al., 2018; Jaimez et al., 2017; Quiroga et al., 2014) for ego-motion prediction from monocular RGB frames. This is achieved by using continuous ego-motion constraints to train a neural network based predictor, which allow the network to remove variations due to depth and moving objects in the input frames (Heeger and Jepson, 1992; Jaegle et al., 2016). Another benefit our approach, in regard to the comparison methods above, is that pixel-wise object-motion can be estimated directly from predicted ego-motion field using flow parsing (Warren and Rushton, 2009).

This chapter is based on a manuscript under review for publication:

Kashyap, H. J., Fowlkes, C., & Krichmar, J. L. (In review). Sparse Representations for Object and Ego-motion Estimation in Dynamic Scenes.

Portions are reprinted.

3.2 Methods

3.2.1 Representation of Ego-motion Using a Sparse Basis Set

We propose to represent ego-motion as depth normalized translation ego-motion field (EMF) and rotational EMF, which can be converted to 6DoF ego-motion parameters in closed form. In this setup, the minimization in Equation 2.6 can be converted to an equivalent regression problem for depth normalized translational EMF and rotational EMF, denoted as ξ_t and ξ_ω respectively. We hypothesize that regression with the EMF constraints from Equation 2.1 will be more robust to variations due to depth and dynamic segments than direct 6DoF ego-motion prediction methods (Zhou et al., 2017; Mahjourian et al., 2018; Yin and Shi, 2018).

Regression of high dimensional output is a difficult problem. However, significant progress has been made using deep neural networks and generative models (Kingma and Ba, 2014; Ranjan et al., 2019; Tung et al., 2017; Vijayanarasimhan et al., 2017). For structured data such as EMF, the complexity of regression can be greatly reduced by expressing the target as a weighted linear combination of basis vectors drawn from a pre-computed dictionary. Then the regression will be a much simpler task of estimating the basis coefficients, which usually has orders of magnitude lower dimension than the target.

Suppose $\tilde{\xi_t}$ is the prediction for depth normalized translational EMF obtained as linear combination of basis vectors from a dictionary T. And $\tilde{\xi_\omega}$ is the prediction for rotational EMF calculated similarly from a dictionary R.

$$\tilde{\xi_t} = \sum_{j=1}^{m} \alpha_j T_j \tag{3.1}$$

$$\tilde{\xi}_{\omega} = \sum_{j=1}^{n} \beta_j R_j \tag{3.2}$$

where α_j and β_j are the coefficients, N is the dimension of the input, and $m, n << N$. Small values of m, n not only lead to computational efficiency, but they also allow each basis vector to be meaningful and generic.

On the other hand, having too few active basis vectors is counterproductive for predictions on unseen data with non-Gaussian variations. For example, PCA finds a small set of uncorrelated basis vectors, however, it requires that the important components of the data have the largest variance. Therefore, in presence of non-Gaussian noise with high variance, the principal components deviate from the target distribution and generalize poorly to unseen data (Choudrey, 2002). Furthermore, a smaller dictionary is more sensitive to corruption of the coefficients due to noisy input.

Therefore, for high dimensional and noisy data, a redundant decomposition of the Equations 3.1, 3.2 is preferred. Dictionaries with linearly dependent bases are called overcomplete and they have been used widely for noise removal applications (Elad and Aharon, 2006; Lewicki and Sejnowski, 2000; Simoncelli et al., 1992) and in signal processing (Donoho et al., 2005; Tseng, 2009). Overcomplete representations are preferred due to flexibility of representation for high dimensional input, robustness, and sparse activation (Lewicki and Sejnowski, 2000). A similar representation was proposed to be used in primary visual cortex in the brain to encode variations in natural scenes (Olshausen and Field, 1997).

To obtain an overcomplete representation, a large set of non-orthogonal bases are required that are more specific than the Euclidean bases, such that each base is used to represent only a small subset of data and only a few bases are activated to represent each datapoint. Finding an overcomplete dictionary can be challenging, because the decomposition is not well defined

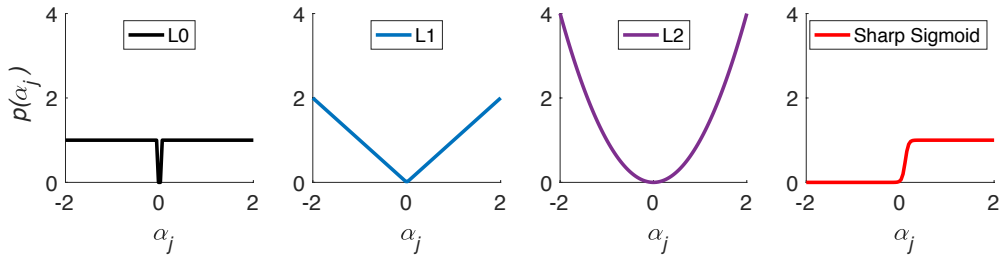


Figure 3.1: L0, L1, and L2 norm penalties and the proposed sharp sigmoid penalty for basis coefficient α_j. It can be observed that for $\alpha_j \geq 0$, the sharp sigmoid penalty approximates the L0 penalty and is continuous and differentiable. The sharp sigmoid function shown above corresponds to $Q = 25$ and $B = 30$. The L1 and L2 norm penalties enforce shrinkage on larger values of α_j. Moreover, for a set of coefficients, L1 and L2 norm penalties cannot indicate the number of $\alpha_j > 0$ due to not having any upper bound.

like in the case of a complete basis set. As pointed out by Lewicki and Sejnowski (2000), despite the flexibility provided by overcompleteness, there is no guarantee that a large set of manually picked linearly dependent basis vectors will fit to the structure of the underlying input distribution (Lewicki and Sejnowski, 2000). Therefore, an overcomplete dictionary must be learned from the data such that the basis vectors encode maximum structure in the distribution. However, this under-determined problem becomes unstable when the input data are inaccurate or noisy (Wohlberg, 2003). Nevertheless, the ill-posedness can be greatly diminished using a sparsity prior on the activations of the basis vectors (Donoho et al., 2005; Elad and Aharon, 2006; Olshausen and Field, 1997). Considering sparse activation prior, the decomposition in Equation 3.1 is constrained by,

$$\|\alpha\|_0 < k \tag{3.3}$$

$\|\alpha\|_0$ is the L0 (pseudo)norm of α and denotes the number of non-zero basis coefficients, with an upper bound k. The decomposition for $\tilde{\xi}_\omega$ in Equation 3.2 is similarly obtained and will not be stated for brevity.

Therefore, the objective function to solve for basis T and co-efficients α can be written as,

$$\underset{T,\alpha}{\operatorname{argmin}} \|\xi_t - \sum_{j=1}^{m} \alpha_j T_j\|_1 \quad \text{subject to} \quad \|\alpha\|_0 < k \tag{3.4}$$

We use L1 norm for the reconstruction error term since it is robust to input noise (Huber, 2004). In contrast, the more commonly used L2 norm overfits to noise, since it results in large errors for outliers (Barnett et al., 1979). As the ξ_t components can be noisy, L1 norm of reconstruction error is more suitable in our case.

The regularizer in Equation 3.4, known as best variable selector (Miller, 2002), requires a pre-determined upper bound k, which may not be the optimal for all samples in a dataset. Therefore, a penalized least squares form is preferred for optimization.

$$\underset{T,\alpha}{\operatorname{argmin}} \|\xi_t - \sum_{j=1}^{m} \alpha_j T_j\|_1 + \lambda_s \|\alpha\|_0 \tag{3.5}$$

The penalty term in Equation 3.5 is computed as $\|\alpha\|_0 = \sum_{j=1}^{m} 1(\alpha_j \neq 0)$, where $1(.)$ is the indicator function. However, the penalty term results in 2^m possible states of the coefficients α and the exponential complexity is not practical for large values of m, as in the case of overcomplete basis (Bertsimas et al., 2016). Further, the penalty function is not differentiable and cannot be solved using gradient based methods.

Although functionally different, the penalty function in Equation 3.5 is commonly approximated using a L1 norm penalty, which is differentiable and results in a computationally

tractable convex optimization problem.

$$\underset{T,\alpha}{\operatorname{argmin}} \|\xi_t - \sum_{j=1}^{m} \alpha_j T_j\|_1 + \lambda_s \|\alpha\|_1 \tag{3.6}$$

Penalized regression of the form in Equation 3.6 is known as Lasso (Tibshirani, 1996), where the penalty $\|\alpha\|_1 = \sum_{j=1}^{m} |\alpha_j|_1$ shrinks the coefficients toward zero and can ideally produce a sparse solution. However, Lasso operates as a biased shrinkage operator as it penalizes larger coefficients more compared to smaller coefficients (Bertsimas et al., 2016; Louizos et al., 2017). As a result, it prefers solutions with many small coefficients than solutions with fewer large coefficients. When input has noise and correlated variables, Lasso results in a large set of activations, all shrunk toward zero, to minimize the reconstruction error (Bertsimas et al., 2016).

To perform best variable selection through a gradient based optimization, we propose to use a penalty function that approximates L0 norm for rectified input based on the generalized logistic function with a high growth rate, which we call as sharp sigmoid penalty and is defined for the basis coefficient α_j as,

$$p(\alpha_j) = \frac{1}{1 + Qe^{-B\alpha_j}} \tag{3.7}$$

where, Q determines the response at $\alpha = 0$ and B determines the growth rate. The Q and B hyperparameters are tuned within a finite range such that i) zero activations are penalized with either zero or a negligible penalty and ii) small magnitude activations are penalized equally as the large magnitude activations (like L0). The sharp sigmoid penalty is continuous and differentiable for all input values, making it a well suited sparsity regularizer for gradient based optimization methods. Thus, the objective function with sharp sigmoid

sparsity penalty can be written as,

$$\underset{T,\alpha}{\operatorname{argmin}} \|\xi_t - \sum_{j=1}^{m} \alpha_j T_j\|_1 + \lambda_s \sum_{j=1}^{m} \frac{1}{1 + Qe^{-B\alpha_j}} \tag{3.8}$$

Figure 3.1 shows that the sharp-sigmoid penalty approximates number of non-zero coefficients in rectified α. It provides a sharper transition between 0 and 1 compared to the sigmoid function and does not require additional shifting and scaling. To achieve dropout like weight regularization (Srivastava et al., 2014), a sigmoid derived hard concrete gate was proposed in (Louizos et al., 2017) to penalize neural network connection weights. However, it does not approximate the number of non-zero weights and averages to the sigmoid function for noisy input.

3.2.2 Joint Optimization for Basis Vectors and Coefficients

We now describe the proposed optimization method to find the basis sets T, R and coefficients α for translational and rotational EMF, based on the objective function in Equation 3.8. We let the optimization determine the coupling between the coefficients for rotation and translation, therefore the coefficients α are shared between T and R. We write the objective in a framework of energy function $E(\xi_t, \xi_\omega | T, R, \alpha)$ as

$$T^*, R^*, \alpha^* = \underset{T,R,\alpha}{\operatorname{argmin}} E(\xi_t, \xi_\omega | T, R, \alpha) \tag{3.9}$$

where

$$E(\xi_t, \xi_\omega | T, R, \alpha) = \lambda_t \|\xi_t - \sum_{j=1}^{m} \alpha_j T_j\|_1 + \lambda_\omega \|\xi_\omega - \sum_{j=1}^{m} \alpha_j R_j\|_1 + \lambda_s \sum_{j=1}^{m} \frac{1}{1 + Qe^{-B\alpha_j}} \quad (3.10)$$

There are three unknown variables T, R, and α to optimize such that the energy in Equation 3.10 is minimal. This can be performed by optimizing over each variable one by one (Olshausen and Field, 1997). For example, expectation maximization procedure can be used to iteratively optimize over each unknown.

For gradient based minimization over α_j, we may iterate until the derivative of $E(\xi_t, \xi_\omega | T, R, \alpha)$ with respect to each α_j is zero. For each input optic flow, the α_j are solved by finding the equilibrium of the differential equation

$$\dot{\alpha_j} = \lambda_t T_j sgn(\xi_t - \sum_{j=1}^{m} \alpha_j T_j) + \lambda_\omega R_j sgn(\xi_\omega - \sum_{j=1}^{m} \alpha_j R_j) - \lambda_s p'(\alpha_j) \quad (3.11)$$

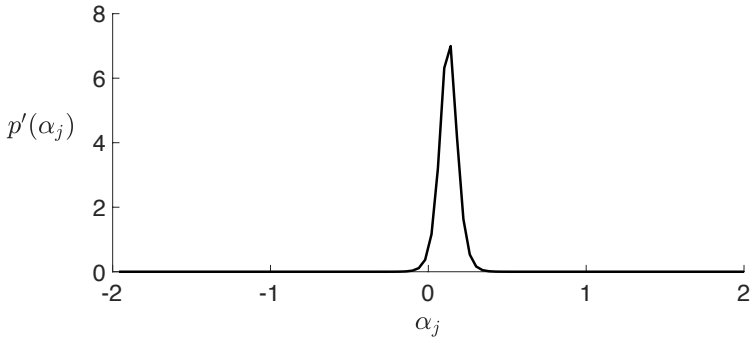


Figure 3.2: Derivative of the sharp sigmoid penalty function $p(\alpha_j)$ with respect to coefficient α_j.

However, the third term of this differential that imposes self-inhibition on α_j is problematic. As depicted in Figure 3.2, the gradient $p'(\alpha_j)$ of the sharp sigmoid penalty with respect to the coefficient is mostly zero, except for a small interval of coefficient values close to zero. As a result, the α_j values outside this interval will have no effect on the minimization to impose sparsity. The sparsity term also has zero derivatives with respect to R and T, therefore

Equation 3.9 cannot be directly optimized over T, R, and α for sparsity when sharp sigmoid penalty is used.

Instead, we can cast it as a parameterized framework where the optimization is solved over a set of parameters θ_s that predicts the sparse coefficients α to minimize the energy form in Equation 3.10. This predictive model can be written as $\alpha = f_{\theta_s}(\hat{v})$. The unknown variables R and T can be grouped along with θ_s as $\theta = \{T, R, \theta_s\}$ and optimized jointly to solve the objective

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, E(\xi_t, \xi_\omega, \alpha|\theta) \tag{3.12}$$

where $E(\xi_t, \xi_\omega, \alpha|\theta)$ is equivalent to the energy function in Equation 3.10, albeit expressed in terms of variable θ.

The objective in Equation 3.12 can be optimized efficiently using an autoencoder neural network with θ_s as its encoder parameters and $\{T, R\}$ as its decoder parameters. The encoder output or bottleneck layer activations provide the basis coefficients α. Following this approach, we propose Sparse Motion Field Encoder (SparseMFE), which learns to predict EMF due to self rotation and translation from optic flow input. The predicted EMF allows direct estimation of 6DoF ego-motion parameters in closed form and prediction of projected object velocities or OMF via flow parsing (Warren and Rushton, 2009).

Figure 3.3 depicts the architecture of the proposed SparseMFE network. The network is an asymmetric autoencoder that has a multi-layer fully convolutional encoder and a single layer linear decoder. We will refer to the Conv1X-4 block at the end of the encoder consisting of $m = 1000$ neurons as the bottleneck layer of the SparseMFE network. The bottleneck layer predicts a latent space embedding of ego-motion from input optic flow. This embedding operates as coefficients α for the basis vectors of dictionaries T and R learned as the fully connected decoder weights. The outputs of all Conv block in the encoder, including the

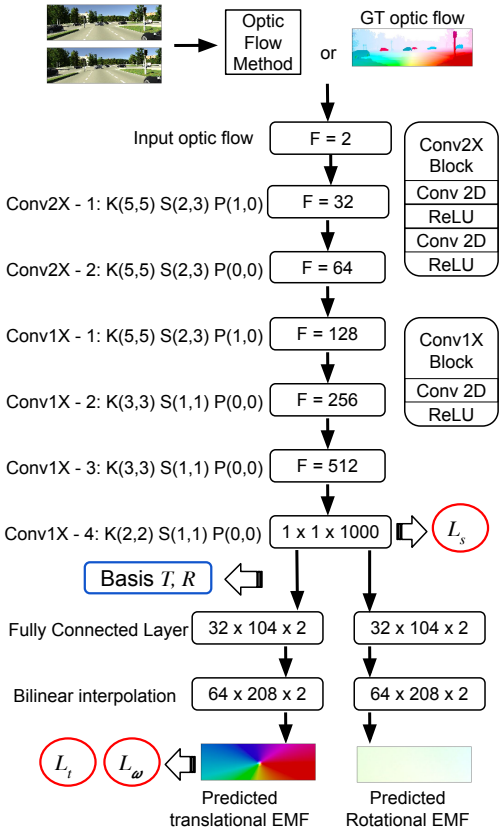


Figure 3.3: Architecture of the proposed SparseMFE network. Conv blocks are fully convolutional layers of 2D convolution and ReLU operations. The receptive field size is gradually increased such that each neuron in the Conv1X-4 layer operates across the entire image. Outputs of all Conv blocks are non-negative due to ReLU operations. K, S, and P denote the kernel sizes, strides, and padding along vertical and horizontal directions of feature maps. F denotes the number of filters in each layer. The weights of the fully connected layer forms the basis for translational and rotational egomotion.

bottleneck layer neurons, are non-negative due to ReLU operations.

EMF reconstruction losses

The translational and rotational EMF reconstruction losses by SparseMFE are obtained as,

$$L_t = \|\xi_t - \tilde{\xi}_t\|_1 \tag{3.13}$$

$$L_\omega = \|\xi_\omega - \tilde{\xi}_\omega\|_1 \tag{3.14}$$

where, ξ_t is true translational EMF with $\rho = 1$ and ξ_ω is true rotational MF, obtained using Equation 2.2.

As most datasets contain disproportionate amount of rotation and translation, we propose to scale L_t and L_ω relative to each other, such that the optimization is unbiased. The scaling coefficients of L_t and L_ω for each input batch are calculated as,

$$\lambda_t = max(\frac{\|\xi_\omega\|_2}{\|\xi_t\|_2}, 1) \tag{3.15}$$

$$\lambda_\omega = max(\frac{\|\xi_t\|_2}{\|\xi_\omega\|_2}, 1) \tag{3.16}$$

Sparsity loss

The SparseMFE network is regularized during training for sparsity of activation of the bottleneck layer neurons. This is implemented by calculating a sparsity loss (L_s) for each batch of data and backpropagating it along with the EMF reconstruction loss during training. The value of L_s is calculated for each batch of data as the number of non-zero activations of the bottleneck layer neurons, also known as population sparsity. Although, to make this loss differentiable, we approximate a number of activations using sharp sigmoid penalty in

Equation 3.7. The penalty L_s is calculated as,

$$L_s = \sum_{j=1}^{m} p(\alpha_j) \tag{3.17}$$

Combining EMF reconstruction loss and sparsity loss, the total loss for training is given by,

$$L = \lambda_t L_t + \lambda_\omega L_\omega + \lambda_s L_s \tag{3.18}$$

where, λ_s is a hyperparameter to scale sparsity loss.

3.3 Experimental results

We evaluate the performance of SparseMFE in ego-motion and object velocity prediction tasks, comapring to the baselines on real KITTI odometry dataset and synthetic MPI Sintel dataset (Geiger et al., 2012; Butler et al., 2012). Additionally, we analyze the EMF basis set learned by SparseMFE for sparsity and overcompleteness.

The predictions for 6DoF translation and rotation parameters are computed in closed form from $\tilde{\xi}_t$ and $\tilde{\xi}_\omega$, respectively, following the continuous ego-motion formulation.

$$\tilde{t} = \tilde{\xi}_t / A \mid \rho = 1, \quad \tilde{\omega} = \tilde{\xi}_\omega / B \tag{3.19}$$

Projected pixel-wise object velocities or OMF are obtained using Equation 2.7.

3.3.1 Datasets

KITTI visual odometry dataset

We use the KITTI visual odometry dataset to evaluate ego-motion prediction performance by the proposed model (Geiger et al., 2012). This dataset provides eleven driving sequences (00-10) with RGB frames (we use only the left camera frames) and the ground truth pose for each frame. Of these eleven sequences, we use sequences 00-08 for training our model and sequences 09, 10 for testing, similar to the related methods (Zhou et al., 2017; Yin and Shi, 2018; Lee and Fowlkes, 2019; Mahjourian et al., 2018). This amounts to approximately 20.4K frames in the training set and 2792 frames in the test set. As ground truth optic flow is not available for this dataset, we use a pretrained PWC-Net model by Sun et al. (2018) to generate optic flow from the pairs of consecutive RGB frames for both training and testing.

MPI Sintel dataset

MPI Sintel dataset contains scenes with fast camera and object movement and also many scenes with large dynamic regions (Butler et al., 2012). Therefore, this is a challenging dataset for ego-motion and OMF prediction. Similar to the other pixel-wise object-motion estimation methods (Lv et al., 2018), we split the dataset such that the test set contains scenes with a different proportion of dynamic regions, in order to study the effect of moving objects on prediction accuracy. Of the 23 scenes in the dataset, we select *alley_2*(1.8%), *temple_2*(5.8%), *market_5*(27.04%), *ambush_6*(38.96%), and *cave_4*(47.10%) sequences as the test set, where the number inside the parentheses specify the percentage of dynamic regions in each sequence. The rest 18 sequences are used to train SparseMFE.

3.3.2 Training

We use Adam optimizer (Kingma and Ba, 2014) to train SparseMFE. Learning rate η is set to 10^{-4} and is chosen empirically by line search. The β_1 and β_2 parameters of Adam are set to 0.99 and 0.999, respectively. The sparsity coefficient λ_s for training is set to 10^2, whose selection criterion is described later in Section 3.3.5. All models are implemented using PyTorch library.

3.3.3 Ego-motion Prediction

For the KITTI visual odometry dataset, following the existing literature on learning based ego-motion prediction (Zhou et al., 2017; Yin and Shi, 2018; Lee and Fowlkes, 2019; Mahjourian et al., 2018; Ranjan et al., 2019), absolute trajectory error (ATE) metric is used for ego-motion evaluation, which measures the distance between the corresponding points of the ground truth and the predicted trajectories. In Table 3.1, we compare the proposed model against the existing methods on the KITTI odometry dataset. Recent deep learning based SfM models for direct 6DoF ego-motion prediction are compared as baselines since their ego-motion prediction method is comparable to SparseMFE. For reference, we also compare against a state-of-the-art visual SLAM method, ORB-SLAM (Mur-Artal et al., 2015) and epipolar geometry based robust optimization methods (Jaegle et al., 2016; Hartley, 1997).

Table 3.1 shows that SparseMFE achieves the state-of-the-art ego-motion prediction accuracy on both test sequences 09 and 10 of the KITTI odometry test split compared to the state-of-the-art learning based ego-motion methods (Yin and Shi, 2018; Mahjourian et al., 2018; Ranjan et al., 2019) and geometric ego-motion estimation baselines (Hartley, 1997; Mur-Artal et al., 2015; Jaegle et al., 2016).

In order to investigate the effectiveness of the learned sparse representation of ego-motion, we

Table 3.1: Absolute Trajectory Error (ATE) on the KITTI visual odometry test set

Method	Seq 09	Seq 10
ORB-SLAM (Mur-Artal et al., 2015)	0.064±0.141	0.064±0.130
Robust ERL (Jaegle et al., 2016)	0.447±0.131	0.309±0.152
8-pt Epipolar + RANSAC (Hartley, 1997)	0.013±0.016	0.011±0.009
Zhou et al. (Zhou et al., 2017)	0.021±0.017	0.020±0.015
Lee and Fowlkes (Lee and Fowlkes, 2019)	0.019±0.014	0.018±0.013
Yin et al. (Yin and Shi, 2018)	0.012±0.007	0.012±0.009
Mahjourian et al. (Mahjourian et al., 2018)	0.013±0.010	0.012±0.011
Godard et al. (Godard et al., 2019)	0.023±0.013	0.018±0.014
Ranjan et al. (Ranjan et al., 2019)	0.012±0.007	0.012±0.008
SparseMFE	**0.011±0.007**	**0.011±0.007**
SparseMFE (top 5% coefficients)	**0.011±0.007**	**0.011±0.007**
SparseMFE (top 3% coefficients)	**0.011±0.007**	**0.011±0.007**
SparseMFE (top 1% coefficients)	0.011±0.008	0.012±0.008

evaluate ATE using only a few top percentile activations of basis coefficients in the bottleneck layer of SparseMFE. This metric tells about dimensionality reduction capabilities of an encoding scheme. As shown in Table 3.1, SparseMFE achieves state-of-the-art ego-motion prediction on both sequences 09 and 10 using only the 3% most active basis coefficients for each input frame pair. Further, when using this subset of coefficients only, the achieved ATE is equal to when using all the basis coefficients. This implies that SparseMFE is able to learn a sparse representation of ego-motion.

On the MPI Sintel dataset, we use the relative pose error (RPE) metric for evaluation of ego-motion prediction (Sturm et al., 2012), similar to the baseline method Rigidity Transform Network (RTN) (Lv et al., 2018). SparseMFE is comparable to this parametric method without any additional iterative refinement of ego-motion. An offline refinement step can be used with SparseMFE as well. However, offline iterative refinement methods are independent of the pose prediction and therefore, cannot be compared directly.

Table 3.2 compares ego-motion prediction performance of SparseMFE against the baseline RTN (Lv et al., 2018), ORB-SLAM (Mur-Artal et al., 2015), geometric ego-motion meth-

Table 3.2: Relative Pose Error (RPE) comparison on the Sintel test set

	dynamic region <10%				dynamic region 10% - 40%				dyn. reg. >40%		All	
	alley_2		temple_2		market_5		ambush_6		cave_4		Average	
	RPE(t)	RPE(r)	RPE(t)	RPE(r)	RPE(t)	RPE(r)	RPE(t)	RPE(r)	RPE(t)	RPE(r)	RPE(t)	RPE(r)
ORB-SLAM (Mur-Artal et al., 2015)	0.030	0.019	0.174	0.022	0.150	0.016	0.055	0.028	**0.017**	0.028	0.089	0.022
Robust ERL (Jaegle et al., 2016)	**0.014**	0.022	0.354	0.019	0.259	0.035	0.119	0.107	0.018	0.046	0.157	0.041
8-pt + RANSAC (Hartley, 1997)	0.058	**0.002**	0.216	**0.006**	0.087	0.012	0.096	0.041	0.018	0.019	0.095	0.013
SRSF (Quiroga et al., 2014)★	0.049	0.014	0.177	0.012	0.157	0.011	0.067	0.073	0.022	0.015	0.098	0.018
VOSF (Jaimez et al., 2017)★	0.104	0.032	**0.101**	0.016	**0.061**	**0.001**	**0.038**	**0.019**	0.044	**0.005**	**0.075**	0.014
RTN (Lv et al., 2018)★	0.035	0.028	0.159	0.012	0.152	0.021	0.046	0.049	0.023	0.021	0.088	0.022
SparseMFE	0.020	0.005	0.172	0.010	0.202	0.011	0.087	0.041	0.025	0.011	0.103	**0.012**

★ denotes that a method uses RGBD input for ego-motion prediction.

ods (Jaegle et al., 2016; Hartley, 1997), and non-parametric baselines SRSF (Quiroga et al., 2014) and VOSF (Jaimez et al., 2017) on the Sintel test split. The lowest and the second lowest RPE on each sequence are denoted using boldface and underline, respectively. ★ denotes that a method uses RGBD input for ego-motion prediction. SparseMSE and the geometric baselines do not use depth input for ego-motion prediction, however, RTN, SRSF, and VOSF use RGBD inputs. For a fair comparison of our method with RTN, both methods obtain optic flow using PWC-net (Sun et al., 2018). SparseMFE achieves the lowest overall rotation prediction error compared to the existing methods, even when using only RGB frames as input. Although, VOSF achieves the lowest overall translation prediction error, it uses depth as an additional input to predict ego-motion (Jaimez et al., 2017).

3.3.4 Object-motion Prediction

We quantitatively and qualitatively evaluate SparseMFE on object-motion prediction using the Sintel test split. We compare to RTN (Lv et al., 2018) and Semantic Rigidity (Wulff et al., 2017) as the state-of-the-art learning based baselines and SRSF (Quiroga et al., 2014) and VOSF (Jaimez et al., 2017) as non-parametric baselines for obejct-motion evaluation. RTN trained using the Things3D dataset (Mayer et al., 2016) for generalization is also included. The standard end-point-error (EPE) metric is used, which measures the euclidean distance between the ground truth and the predicted 2D flow vectors generated by moving objects. These 2D object flow vectors are herein referred to as OMF and with a different terminology "projected scene flow" by Lv et al. (2018). Table 3.3 shows that SparseMFE achieves the state-of-the-art OMF prediction accuracy on four out of five test sequences. The lowest EPE per sequence is denoted in boldface. The other methods become progressively inaccurate with larger dynamic regions. On the other hand, SparseMFE maintains OMF prediction accuracy even when more than 40% of the scene is occupied by moving objects, as in case of the *cave_4* sequence.

Table 3.3: End Point Error (EPE) comparison of OMF prediction on the Sintel test split

	dynamic region <10%		dynamic region 10% - 40%		dynamic region >40%	All
	alley_2	temple_2	market_5	ambush_6	cave_4	Average
SRSF (Quiroga et al., 2014)	7.78	15.51	31.29	39.08	13.29	18.86
VOSF (Jaimez et al., 2017)	1.54	8.91	35.17	24.02	9.28	14.61
Semantic Rigidity (Wulff et al., 2017)	0.48	5.19	13.02	19.11	6.50	7.39
RTN (trained on Things3D (Mayer et al., 2016))	0.52	9.82	16.99	52.21	5.07	11.88
RTN (Lv et al., 2018)	0.48	**3.27**	11.35	19.08	4.75	6.12
SparseMFE	**0.29**	4.59	**11.27**	**4.82**	**0.93**	**4.32**

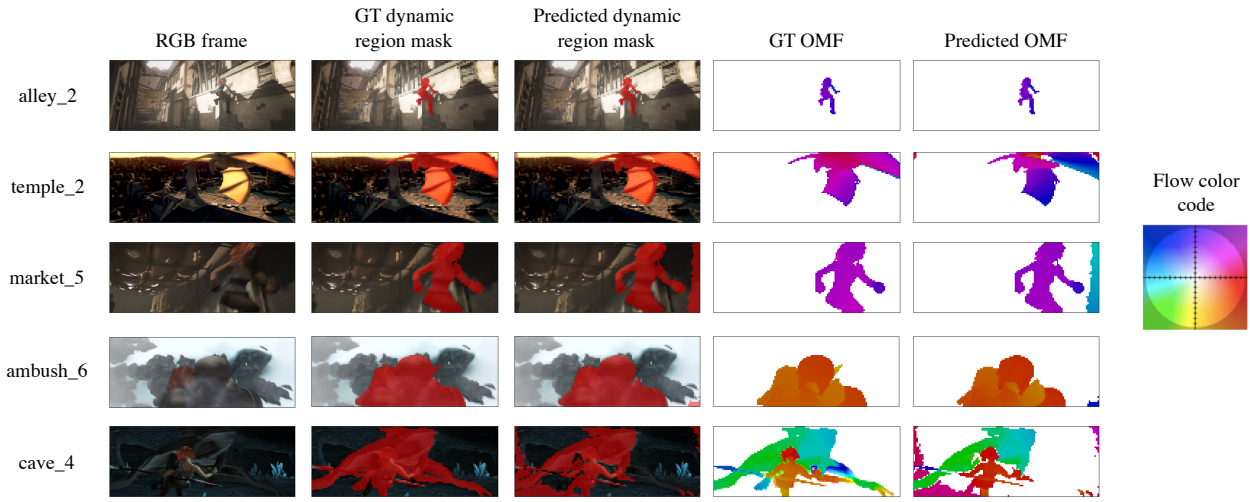


Figure 3.4: Qualitative results of SparseMFE on Sintel test split. The red colored overlay denotes the dynamic region masks.

Figure 3.4 depicts qualitative OMF performance of SparseMFE on each of the five sequences from the Sintel test split. Dynamic region mask is obtained by thresholding the residual optic flow from Equation 2.7. While SparseMFE successfully recovers OMF for fast moving objects, it is possible that some rigid background pixels with faster flow components are classified as dynamic regions, as for the examples from *market_5* and *cave_4* sequences. This can be avoided by using more data for training, since these background residual flows are generalization errors stemming from ego-motion prediction and are absent in training set predictions.

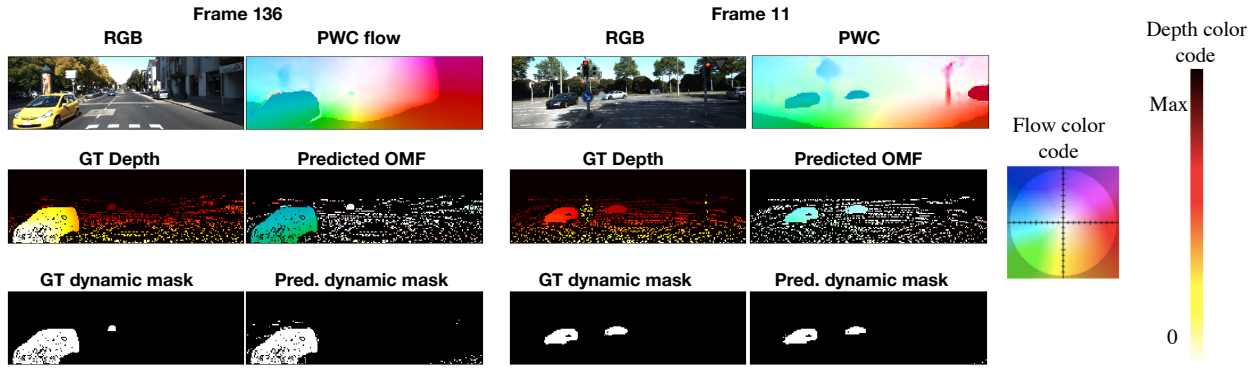


Figure 3.5: Qualitative results of SparseMFE on KITTI benchmark real world frames (Menze and Geiger, 2015). Ground truth OMF is not available, however, ground truth dynamic region masks are provided in the benchmark. The ground truth depth map is sparse, and the pixels where depth is not available are colored in black.

We show qualitative object-motion prediction results on real world KITTI benchmark by Menze and Geiger (2015) in Figure 3.5, which illustrates effective dynamic region prediction compared to ground truth dynamic region masks. The benchmark does not provide ground truth OMF, which are difficult to obtain for real world scenes.

3.3.5 Sparsity Analysis

We analyze the effects of using the sparsity regularizer for encoding ego-motion. The proposed sharp sigmoid penalty in Equation 3.7 is compared against L1 and L2 norm sparsity penalties commonly used in sparse feature learning methods (Jiang et al., 2015; Hoyer, 2004). ReLU non-linearity at the bottleneck layer was proposed for sparse activations by Glorot et al. (2011). Since the bottleneck layer of Sparse MFE uses ReLU non-linearity, we also compare the case where no sparsity penalty is applied.

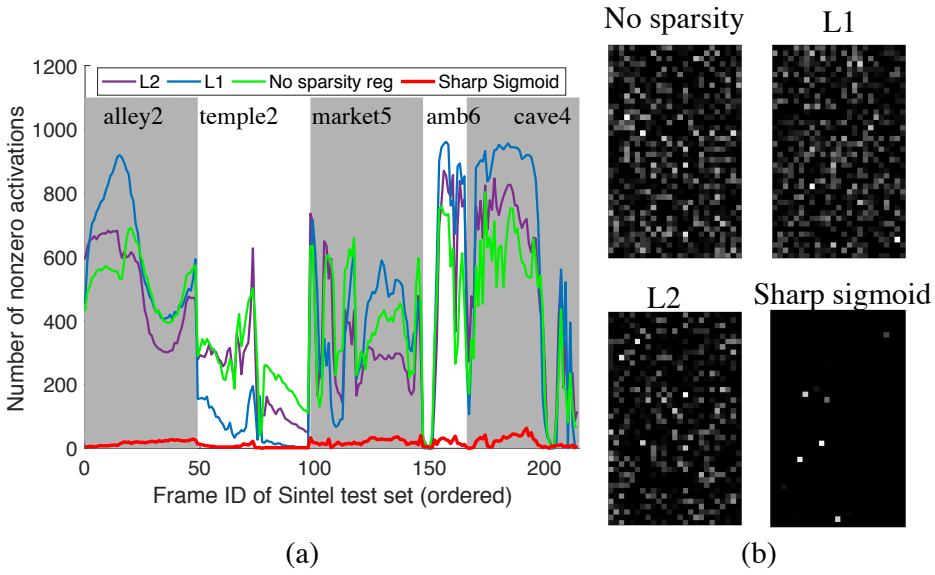


Figure 3.6: Neuron activation profile in the bottleneck layer on Sintel test split for different types of sparsity regularization. (a) Number of nonzero activations in the bottleneck layer for frame sequences in the Sintel test split. Line colors denote the sparsity regularization used. (b) Activation heatmap of the bottleneck for the *market_5* frame shown in Figure 3.4. All experiments are conducted after the network has converged to a stable solution.

Figure 3.6 depicts the effectiveness of the proposed sharp sigmoid penalty in learning a sparsely activated basis set for ego-motion prediction. Figure 3.6(a) shows the number of nonzero activations in the bottleneck layer on Sintel test split when the network is trained using different sparsity penalties. Sharp sigmoid penalty results in sparse and stable activations of basis coefficients for all Sintel test sequences. On the contrary, L0 and L1 norm penalties find dense solutions where large basis subsets are used for all sequences. Figure 3.6(b) shows the activation heatmap of the bottleneck layer for the *market_5* frame in Figure 3.4 for the tested sparsity penalties. L0 and L1 penalties do not translate to the number of nonzero activations, rather work as a shrinkage operator on activation magnitude, to result in large number of small activations in the bottleneck layer. On the other hand, the proposed sharp sigmoid penalty activates only a few neurons in that layer.

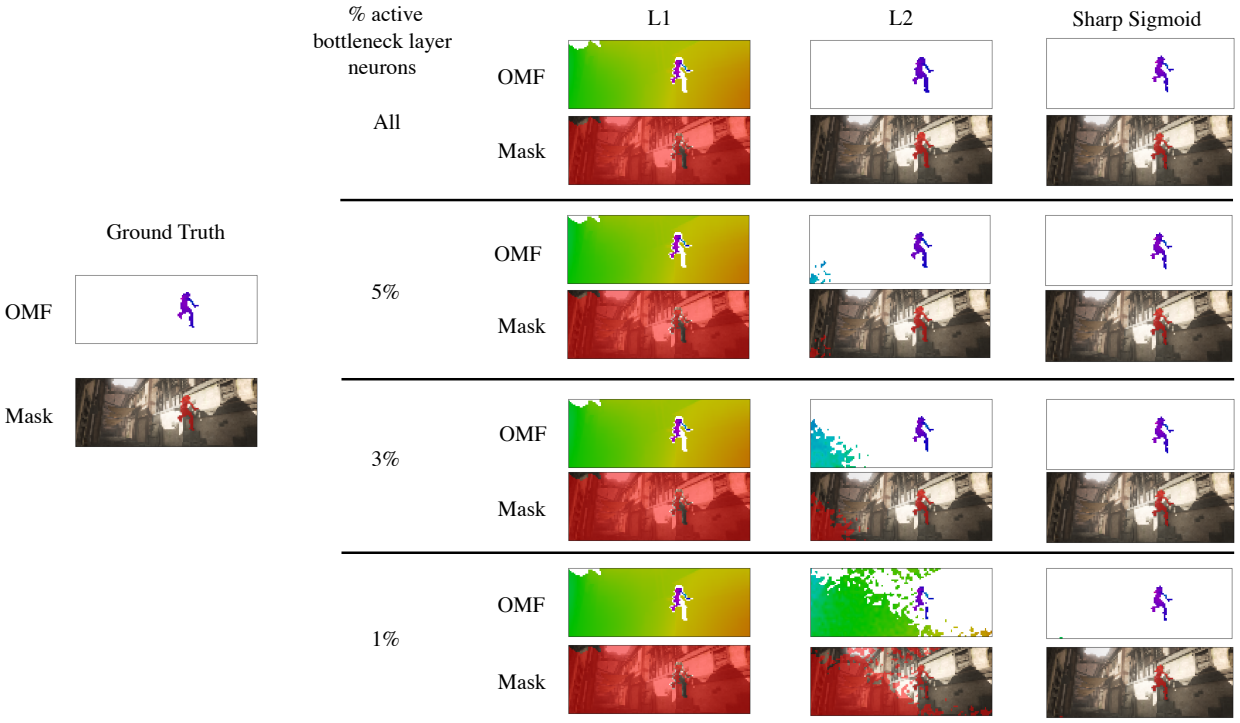


Figure 3.7: Qualitative OMF and dynamic mask prediction results comparing L1, L2, and Sharp Sigmoid sparsity penalties, in terms of their robustness to removal of bottleneck layer neurons during testing.

We conducted ablation experiments to study the effectiveness of L1, L2, and sharp sigmoid penalties in learning a sparse representation of ego-motion. To do so, we observed

the performance of SparseMFE trained using either L1, L2, or sharp sigmoid penalties after removal of neurons in the bottleneck layer during test. Figure 3.7 depicts qualitative OMF and dynamic mask prediction performance on the *alley_2* test frame from Figure 3.4 by SparseMFE instances trained using either L1, L2, or sharp sigmoid penalties, with or without ablation. During ablation, we use only a fraction of the top bottleneck neuron activations (coefficients) and set the others to zero. The results show that sharp sigmoid penalty based training provides stable OMF and dynamic mask prediction using only top 1% largest activations, whereas L2 sparsity penalty based training results in loss of accuracy as neurons are removed from bottleneck layer. L1 penalty based training results in erroneous OMF and mask predictions for this example.

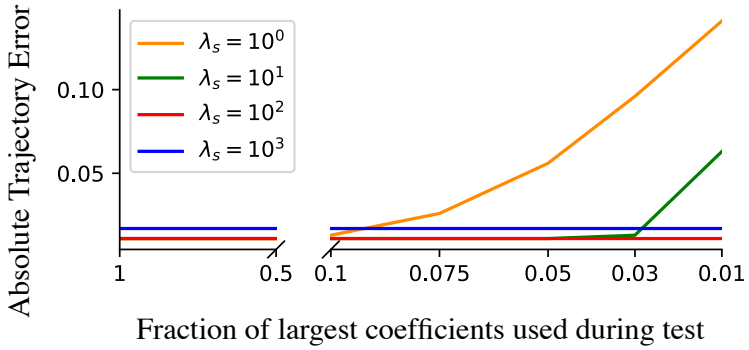


Figure 3.8: Ablation experiment to study the effect of the sparsity loss coefficient λ_s on ego-motion prediction. During test, only a fraction of the bottleneck layer neurons are used for ego-motion prediction based on activation magnitude and the rest are set to zero. ATE is averaged over all frames in KITTI test sequences 09 and 10.

To study the effect of the sparsity loss coefficient λ_s on ego-motion prediction, we conducted a study by varying λ_s during training and using only a fraction of the most activated bottleneck layer neurons for ego-motion prediction during test and setting the rest to zero. Figure 3.8 depicts the effect of ablation on the ego-motion prediction accuracy during test, for λ_s values in the set $\{10^e | 0 \leq e < 4, e \in \mathbb{Z}\}$. As can be seen, $\lambda_s = 10^2$ achieves the smallest and stable ATE for different amount of ablation. For smaller λ_s values, the prediction becomes inaccurate as more bottleneck layer neurons are removed. Although $\lambda_s = 10^3$ provides stable prediction, it is less accurate than $\lambda_s = 10^2$. The stability to ablation of neurons for larger

λ_s values is a further indication of the effectiveness of the sharp sigmoid sparsity penalty in learning a sparse basis set of ego-motion.

3.3.6 The Learned Basis Set

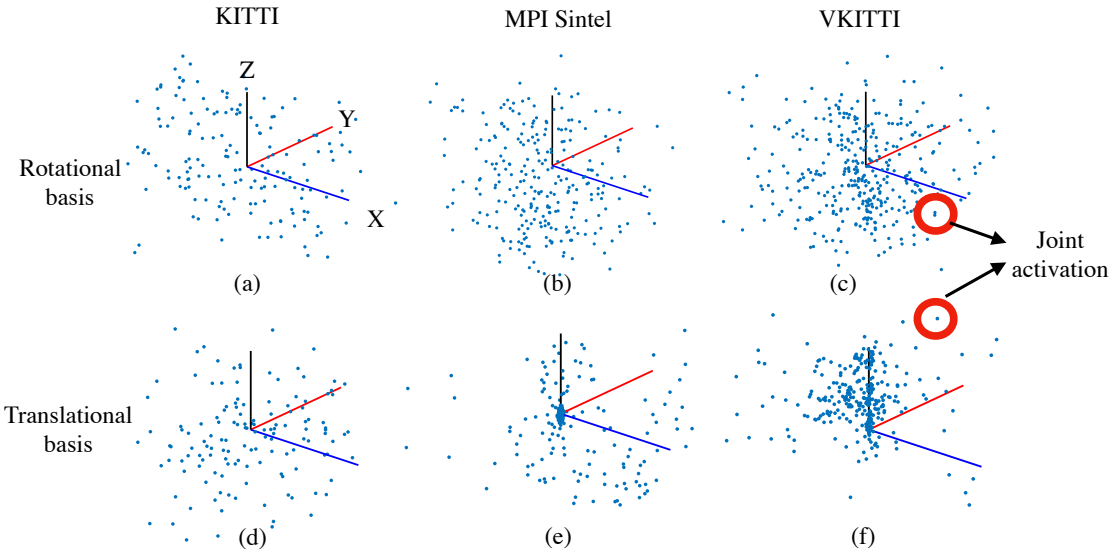


Figure 3.9: Projection of the learned EMF basis set for rotational and translational ego-motion to the Euclidean space in the camera reference frame. The dots represent the learned bases and the solid lines represent the positive X, Y, and Z axes of the Euclidean space. The red circles indicate a pair of translation and rotation bases that share a same coefficient.

We visualize the EMF basis sets R and T learned by SparseMFE in Figure 3.9 by projecting them onto the three dimensional Euclidean space in the camera reference frame using Equation 3.19. We also include the basis sets learned from training on another synthetic dataset VKITTI (Gaidon et al., 2016). It can be seen that the learned R and T are overcomplete, i.e. redundant and linearly dependent (Lewicki and Sejnowski, 2000; Olshausen and Field, 1997). The redundancy helps in two ways, first, to use different basis subsets to encode similar ego-motion so that the individual bases are not always active. Second, if some basis subsets are turned off or get corrupted by noise, the overall prediction is still robust (Lewicki and Sejnowski, 2000; Simoncelli et al., 1992). Moreover, a pair of translational and rotational bases share the same coefficient to encode ego-motion. In that sense, the bottleneck layer

neurons are analogous to the parietal cortex neurons of the primate brain that jointly encode self rotation and translation (Sunkara et al., 2016).

An observation from Figure 3.9 is that the learned basis sets can be skewed if the training dataset does not contain enough ego-motion variations. In most sequences of the VKITTI dataset, the camera mostly moves with forward translation (positive Z axis). The learned translation basis set from VKITTI dataset in Figure 3.9(f) shows that most bases lie in the positive Z region, denoting forward translation. Although the KITTI dataset has similar translation bias, we augment the dataset with backward sequences. As a result, the translation basis set learned from the KITTI dataset does not have a skew toward forward translation, as shown in Figure 3.9(d).

3.4 Discussion

In this chapter, we introduce a novel approach for predicting 6DoF ego-motion and image velocity generated by moving objects from optic flow input. In particular, our approach considers motion-field decomposition in terms of ego and object-motion sources in the dynamic image segments. This is in contrast to existing ego-motion estimation approaches that mask the dynamic image segments for direct 6DoF egomotion estimation (Zhou et al., 2017; Lee and Fowlkes, 2019; Vijayanarasimhan et al., 2017) or other approaches that assume the scene to be static (Hartley, 1997; Zhang and Tomasi, 2002; Fredriksson et al., 2015). Our approach predicts the ego-motion field covering both rigid background and dynamic segments, from which object-motion and 6DoF ego-motion parameters can be derived in closed form.

To achieve robust ego-motion field prediction in the presence of variations due to depth and moving objects, an overcomplete sparse basis set of rotational and translational ego-motion is learned using a convolutional autoencoder with a nonzero basis activation penalty at the

bottleneck layer. The proposed asymmetric autoencoder has a single layer linear decoder that learns the translational and rotational ego-motion basis sets as connection weights, whereas a fully convolutional encoder provides the basis coefficients that are sparsely activated. We show that L1 and L2 norm penalties do not lead to sparse activations of the ego-motion bases (Figure 3.6). Although L0 norm estimates the number of nonzero activations and is a suitable penalty term for sparsity regularization, it is not continuous and differentiable. In order to penalize the number of non-zero neuron activations at the bottleneck layer during backpropagation training, we propose a continuous and differentiable sparsity penalty term that approximates L0 norm for rectified signal, such as ReLU activation output. Compared to the L1 norm and L2 norm penalties, the proposed sparsity penalty is advantageous since it penalizes similar to the uniform L0 norm operator and does not result in a large number of activations.

Experiments conducted using the real world KITTI odometry dataset Geiger et al. (2012) indicate that SparseMFE achieves state-of-the-art ego-motion prediction accuracy in regard to the existing baselines for ego-motion prediction comprising both learning based and geometric methods (Zhou et al., 2017; Ranjan et al., 2019; Yin and Shi, 2018; Mahjourian et al., 2018; Hartley, 1997). The baseline methods predict 6DoF egomotion parameters directly from input, as opposed to our approach of motion field prediction. These results demonstrate that the proposed motion field predictor trained using the continuous egomotion constraints generalizes effectively to the test sequences (Table 3.1).

An additional benefit of our approach is that object motion field can be estimated directly from optic flow using flow parsing (Equation 2.7). On the synthetic MPI Sintel dataset with naturalistic first order image and motion statistics (Butler et al., 2012), SparseMFE achieves state-of-the-art object-motion prediction accuracy compared to the existing baselines (Lv et al., 2018; Quiroga et al., 2014; Jaimez et al., 2017). In order to evaluate the effect of the larger moving objects on prediction accuracy, we tested on sequences where moving

objects occupy different ranges of pixels: less than 10%, between 10% and 40%, and greater than 40%, similar to Lv et al. (2018). In contrast to the baseline methods that are more inaccurate for sequences with larger dynamic segments, SparseMFE prediction is robust to this variation (Table 3.3).

The ego-motion basis set learned by SparseMFE is overcomplete and the bases are linearly dependent (Figure 3.9). For large dictionaries, using too many bases to represent each data instance could result in high dimensionality. In such cases, sparse activation of the bases are necessary to reduce the dimension of encoding. SparseMFE achieves state-of-the-art ego-motion prediction accuracy on KITTI odometry dataset using only the 3% most active basis coefficients, with all other coefficients set to zero (Table 3.1). Therefore, the proposed sparsity regularization effectively optimizes for reduced dimensionality of the overcomplete ego-motion representation, even in presence of depth and object-motion variations. In comparison, although PCA finds uncorrelated bases for optimal dimensionality reduction, it cannot be used to identify the underlying distribution in presence of independent non-Gaussian noise (Choudrey, 2002). In this case, PCA cannot be used learn ego-motion basis from optic flow in presence of unconstrained scene depth and object movement.

Finally, as shown in Figure 3.6, the sharp sigmoid sparsity penalty proposed herein is more effective in enforcing sparsity on the basis coefficients, compared to L1 and L2 norm based sparsity penalties used in popular regularization approaches Lasso and ridge regression, respectively (Tibshirani, 1996; Hoerl and Kennard, 1970). L1 and L2 norm penalties work as shrinkage operators on the coefficient values (Figure 3.1). Minimizing over these functions result in small values of the basis coefficients that are not zero. On the other hand, minimizing the sharp sigmoid penalty results in fewer nonzero basis coefficients. Also, the resulting representation is more robust to ablation of coefficients (Figure 3.7). Although our experiments consider motion estimation in videos, the regularization techniques developed are also applicable to sparse feature learning from other high dimensional data.

Chapter 4

Convolutional Neural Network Model of Cortical Visual Motion Perception

4.1 Introduction

The sparse motion estimation method proposed in Chapter 3 solved the problem of object-motion estimation from a frame sequence analytically, however it did not suggest a neural mechanism that can solve for both object and ego-motion. A common neuron population that encodes both object and ego-motion makes intuitive sense, since these problems are intertwined. Moreover, in the motion estimation method presented in Chapter 3, the depth input is externally provided during flow parsing for object motion extraction (Equation 2.7). For a neuron population to be able to estimate object motion, it must combine depth information to account for the correct scale of apparent motion on the retina or image sensor due to self-translation (Heeger and Jepson, 1992).

As a reference, we can try to understand the computational mechanisms used by the neurons in MST area of the visual cortex in the brain, where neurons have been found to be selective

for range of object and ego-motion visual input (Sato et al., 2010; Duffy and Wurtz, 1991a,b; Komatsu and Wurtz, 1988). Both dorsal and lateral subdivisions of MST, viz., MSTd and MSTl, have been found to combine visual cues about object and ego-motion (Sasaki et al., 2017, 2019; Eifuku and Wurtz, 1998). Their responses may contribute to both observer heading estimate (Takahashi et al., 2007; Froehler and Duffy, 2002) and figure-ground separation for moving objects (Recanzone et al., 1997; Geesaman and Andersen, 1996). Studies are starting to show how these areas might combine depth and motion cues during ego-motion (Yang et al., 2011; Layton and Niehorster, 2019). Also, human experiments have shown that depth is essential to estimating the direction of moving objects (Matsumiya and Ando, 2009; Warren and Rushton, 2007).

In this chapter, we present a computational model of object and ego-motion perception, that combines visual cues about motion and depth in the same neural population. To benefit from the repetitive patterns in optic flow and to achieve generalization to new stimuli, we use multiple convolutional neuron layers in the model, which have been highly successful in learning complex patterns from visual data (Krizhevsky et al., 2012; LeCun et al., 1998). We train our convolutional neural network on object and ego-motion estimation tasks without any prior assumption about neuron activation behaviors. We find that the hidden neuron layer representations resulting from the goal driven training are comparable to MSTd neuron responses for identical optic flow, object motion, and combined stimuli (Sato et al., 2010).

4.2 Methods

4.2.1 Convolutional Neural Network Model

The architecture of the convolutional neural network (CNN) model (LeCun et al., 1998; Krizhevsky et al., 2012) used in this study is depicted in Figure 4.1. The network receives

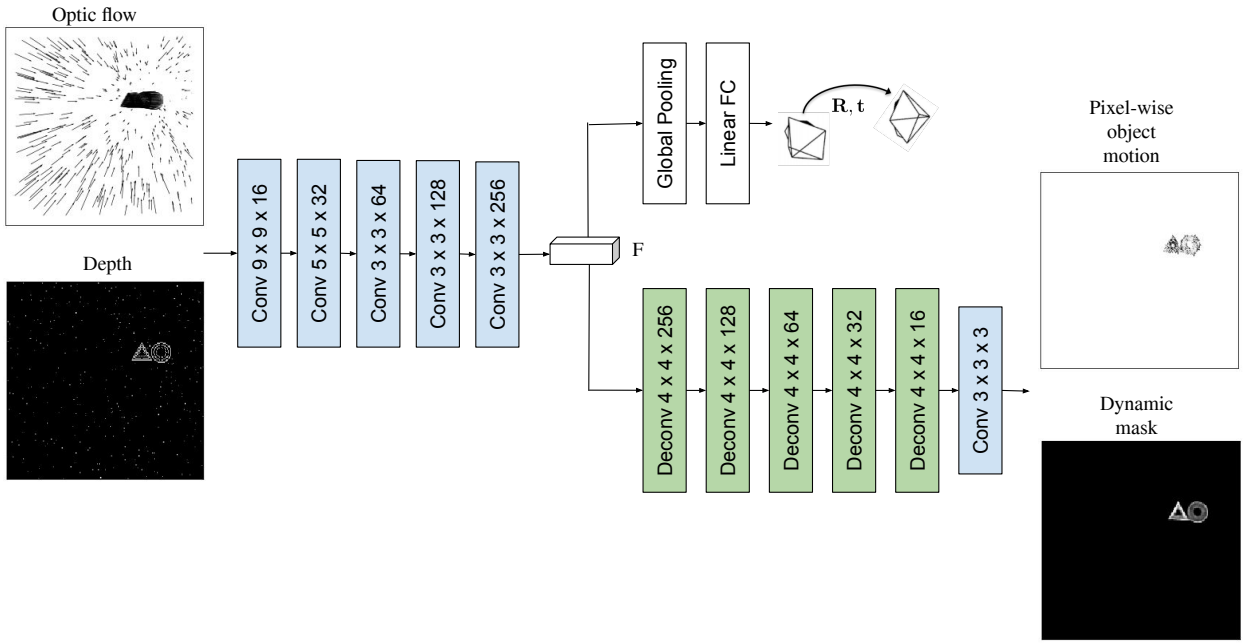


Figure 4.1: The convolutional neural network used to simulate the MSTd-like model neurons that respond to interactions between object motion and optic flow. The network receives 2D optic flow and pixel-wise depth map as input and predicts 6DoF ego-motion parameters, pixel-wise 2D object motion, and dynamic object mask. The example in the figure shows a stimulus made of counter-clockwise observer translation along the horizontal plane with speed $3.14m/s$, observer rotation $1°/s$ w.r.to the vertical axis, and simultaneous clock-wise object motion ($3.14m/s$). The convolutional layers output a feature activation matrix F, which is used to decode the three outputs.

two inputs, i) 2D optic flow caused by observer translation, rotation and independent object movements and ii) pixel-wise depthmap of the scene. We represented input in this way to focus on the higher order motion processing mechanisms of the dorsal visual pathway. They could be seen as the output of the early visual processing stages that extract local motion from input images (e.g. LGN and V1) and perform binocular stereo matching (e.g. V1 and V2) (Layton and Fajen, 2020). The network consists of an encoder with five convolutional layers with kernel sizes $\{9 \times 9,\ 5 \times 5,\ 3 \times 3,\ 3 \times 3,\ 3 \times 3\}$ and feature map sizes $\{16, 32, 64, 128, 256\}$. The output of the final encoder layer is a feature activation matrix F of size $7 \times 7 \times 256$. F is used as input to two separate decoder networks, one predicts 6DoF observer rotation and translation parameters through linear transformations and the

other predicts pixel-wise object motion and soft dynamic object mask, using five transposed convolution operations (denoted as Deconv in Figure 4.1), each with kernel sizes 4×4 and a convolution layer with kernel size 3×3. A sigmoid function is used to predict the soft dynamic object mask.

The network is trained end-to-end using backpropagation (Rumelhart et al., 1995). The total loss for training is calculated by combining the prediction errors, viz., self-translation loss (L_t), self-rotation loss (L_ω), object motion loss (L_o), and dynamic mask loss (L_m), using homoscedastic multitask loss scaling (Kendall et al., 2018), and is derived as,

$$L_{total} = \frac{1}{2\sigma_t^2} L_t + \frac{1}{2\sigma_\omega^2} L_\omega + \frac{1}{2\sigma_o^2} L_o + \frac{1}{\sigma_m^2} L_m + \lambda_s \sum_{j=1}^{|F|} p(\alpha_j) + log(\sigma_t \sigma_\omega \sigma_o) + log(\sigma_m) \quad (4.1)$$

where,

$$L_t = \left\| t - \tilde{t} \right\|_2 \quad (4.2)$$

$$L_\omega = \left\| \omega - \tilde{\omega} \right\|_2 \quad (4.3)$$

$$L_o = \left\| f_o - \tilde{f}_o \right\|_2 \quad (4.4)$$

$$L_m = -\sum_{i=1}^{N} m_i \log(\tilde{m}_i) + (1 - m_i) \log(1 - \tilde{m}_i) \quad (4.5)$$

t is 3D self-translation velocity, ω is 3D self-rotation velocity, f_o is pixel-wise object motion, m is the dynamic object mask, and the corresponding terms with a tilde($\tilde{}$) sign on top are their predictions by the network. N is number of pixels in the input depthmap and optic flow. σ_o, σ_t, σ_ω, and σ_m are learnable homoscedastic loss scaling coefficients. Prediction errors for object motion and ego-motion are calculated using L2 norm and error of mask prediction is calculated using binary cross entropy loss. A sparsity penalty of the activations

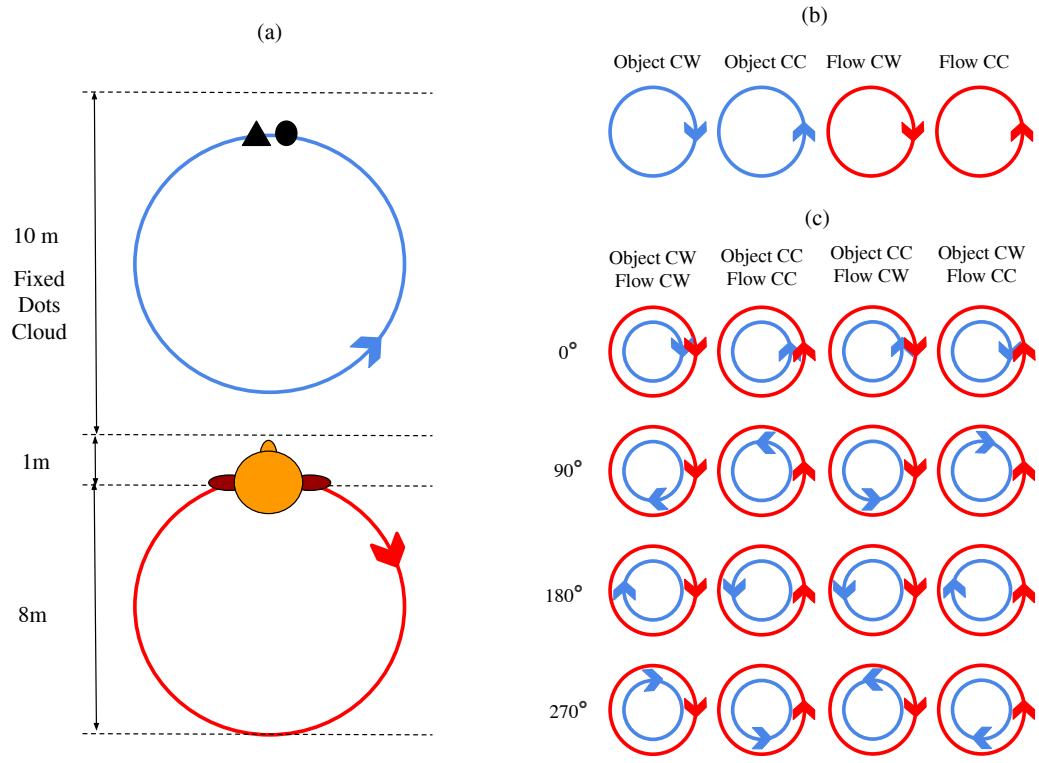


Figure 4.2: Configuration of the virtual scene and the motion patterns used to generate visual stimuli (adapted from (Sato et al., 2010)). (a) Optic flow and object motion visual stimuli simulate translational movement of the observer along the red circular trajectory of 8 meter diameter, in front of earth fixed cloud of dots. The two-part object consists of a triangle and circle and moves along the blue circular trajectory to simulate self-motion. (b) The stimulus conditions containing either object motion or optic flow in clockwise or anticlockwise direction. (c) Sixteen stimulus conditions combining four clockwise/anticlockwise direction combinations with four phase rotations between optic flow and object motion. The locations of the arrows denote the initial positions of the observer and object derived from the phase difference between optic flow and object motion.

in F is used as a regularizer during training. α_j is the activation of the j-th neuron in F, p is the sparsity penalty function given by Equation 3.7 in Chapter 3, and λ_s is a user defined sparsity loss coefficient.

4.2.2 Visual Stimuli

For evaluation of the model in regard to the MSTd neurons, we mimic the stimulus conditions used for neuronal response analysis by Sato et al. (2010). Figure 4.2 depicts the visual stimuli used to tested the model under the same conditions as (Sato et al., 2010). Optic flow and depth input to the model were generated by simulating the apparent motion on the retina resulting from translation of the observer along a horizontal circular path of radius 8 m in cycles of 8 s, in front of a cloud of 1000 white dots with a black background. The 3D dot volume is sampled randomly in each stimulus condition. We used the motion field model by Longuet-Higgins and Prazdny (1980) to generate retinal motion vectors in response to observer self-motion, given by Equation 2.2.

Object motion is simulated using circular motion of 8 s periodic cycle of a two part object consisting of a triangle and a circle within the dot clouds. The movement of the object projects additional local motion on the retina. The triangle object-part is constructed using three concentric triangles with the sides constructed using 130 white dots. Similarly, the circle object-part is constructed using three concentric circles with the periphery constructed using 120 white dots. Both object parts are transparent and the static white dots simulating the background are visible through the objects.

Three categories of stimulus condition are used to probe the activation interactions in response to optic flow and object motion. In the first category, only optic flow simulated by the movement of background dots due to observer self-motion in clockwise/counterclockwise (CW/CC) directions are presented (Figure 4.2b, red circles). Second, only retinal motion projections created by the object moving in CW/CC directions are presented (Figure 4.2b, blue circles). And in the third category, sixteen combinations of optic flow and object motion are presented varied by four relative phases from $\{0°, 90°, 180°, 270°\}$ and four relative CW/CC directions. We term them as combined stimuli conditions. For the congruent con-

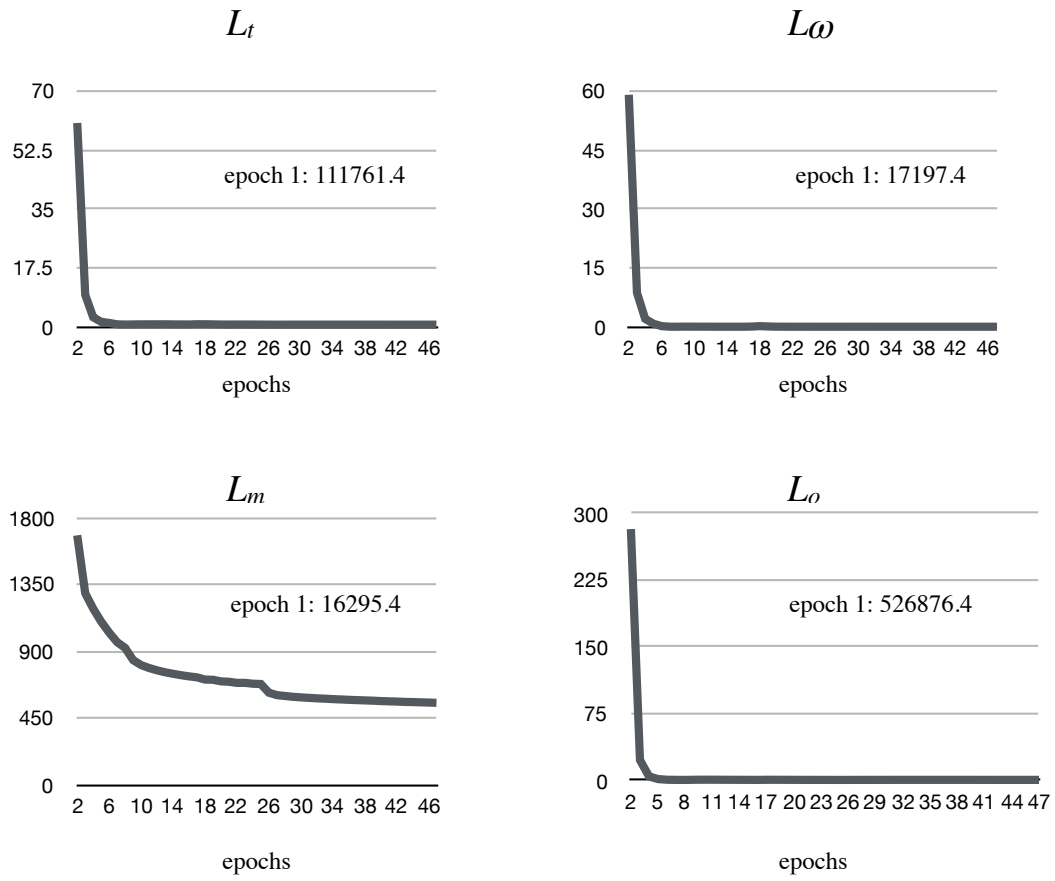


Figure 4.3: The loss terms averaged over all training samples during different epochs of training. Loss values are represented in arbitrary units.

ditions of CW-CW and CC-CC directions between optic flow and object motion, the object dots moves as a part of the static background (Figure 4.2c). In the CW-CC and CC-CW conditions, the object moves independently.

4.2.3 Training

To train the CNN, we created a separate dataset consisting of 27240 frames with additional naturalistic viewing conditions, such as, observer self-rotations between $[-10°/s, 10°/s]$, randomly selected relative phase rotations between object motion and optic flow $[0°, 360°]$, elevated self-motion trajectories sampled uniformly from the range of all possible elevations,

and also no self-motion condition for 10% random frames. These are included to prevent the network from overfitting to a limited set of visual stimuli and to improve generalization.

The network is trained using Adam optimizer (Kingma and Ba, 2014) with a batch size of 4. We use a gradually deceasing learning rate from $2e^{-4}$ to $1e^{-5}$. On the other hand, the sparsity coefficient λ_s is gradually increased from $1e^{-4}$ to $1e^{-2}$ during training. These hyper-parameters are selected empirically based on prediction accuracy and sparsity of activations in F. Figure 4.3 depicts the average training losses during training epochs. The loss values for the first training epoch are stated numerically due to their large magnitude. Training was stopped when improvement of all loss terms saturated.

4.2.4 Neuron Activation Analysis

Model Neuron Selection Protocol

We hypothesize that the activations of features in F, which are outputs of the last convolution layer (conv5) of the encoder, are analogous to the MSTd neuron responses reported by Sato et al. (2010). To evaluate this hypothesis, we compare the activations of the conv5 neurons to the MSTd neuron responses for the same set of visual stimuli. Sato et al. (2010) found that out of their 61 MSTd neurons reported, most neurons fit a a linear regression model (with p-values < 0.05) that described the responses to combined stimuli as a function of the responses to corresponding component object motion and optic flow stimuli. We use that as a criteria to select conv5 neurons for response interaction analysis that corresponds to the MSTd neurons. For each conv5 neuron, we fit a multiple linear regression model of its responses to combined stimuli in terms of its responses to the component object motion and optic flow stimuli across all stimulus conditions. The conv5 neurons that yielded significant models (p-values < 0.05) with a valid R^2 value of fit are considered. Of the actual number of conv5 neurons conv5 neurons, 318 neurons are considered with this criteria and we refer

to them as MSTd-like model neurons.

Baseline Activity Subtraction

Sato et al. (2010) reported that the baseline activity of MSTd neurons, in the absence of visual stimuli, skewed the response to stimuli. They subtracted the baseline activity from the MSTd responses to find more specific nature of additivity pattern across the neuron population. Since the implications of baseline activations in CNNs are not known, we investigate their existence and significance by performing two types of baseline subtractions from raw neuron activation, viz., average and blank baseline subtraction. In case of average baseline subtraction, we subtract the average activation of the neuron across all relative phases and direction between object motion and optic flow, sampled from sets P and D, respectively.

$$\bar{\alpha}_j(s_{pd}(i)) = \alpha_j(s_{pd}(i)) - \frac{1}{|P||D|T} \sum_{p \in P} \sum_{d \in D} \sum_{i=1}^{T} \alpha_j(s_{pd}(i)) \tag{4.6}$$

where, $\alpha_j(s_{pd}(i))$ is the activation of the j-th MSTd-like model neuron in response to i-th input stimulus with phase rotation p and relative direction d between object motion and optic flow. T is the duration of each stimulus condition.

In case of blank baseline subtraction, we subtract the average response of the neuron to the presentation of a blank stimulus (ϕ) without fixed dots and object.

$$\hat{\alpha}_j(s_{pd}(i)) = \alpha_j(s_{pd}(i)) - \frac{1}{T} \sum_{i=1}^{T} \alpha_j(\phi) \tag{4.7}$$

Categorization of Neuron Stimulus Preference

In order to judge if a neuron is highly selective to a particular category of visual stimulus compared to the other categories, we compare its maximal activations to each category of stimulus. A neuron j is assigned to have a preference for stimulus category C_k based on the condition stated below.

$$n \in C_k \Leftrightarrow max(\alpha_j(s(i)),\, \forall i \in C_k) > max(T_c,\, R_c * max(\alpha_j(s(i)),\, \forall i \in C_{k' \neq k})) \qquad (4.8)$$

where, T_c and R_c parameters define the threshold absolute activation value and the minimum ratio to the maximum activation of neuron j for other stimulus categories $C_{k' \neq k}$, respectively, required for assignment of preference to category C_k.

4.3 Results

4.3.1 Neuronal Response Additivity

We examined the effects of combining optic flow and object motion stimuli on the responses of MStd-like model neuron population in comparison to presentation of object motion and optic flow stimuli alone, and compared our results to the neurophysiological data by Sato et al. (2010). For each of the sixteen combined stimulus conditions and the four single stimulus conditions (Section 4.2), we divided model neuron activations during the 8 s stimulus presentation to 40 response intervals of 200 ms each. For each interval of each condition, the activation was normalized to the peak activation for each model neuron. Also, the sum of

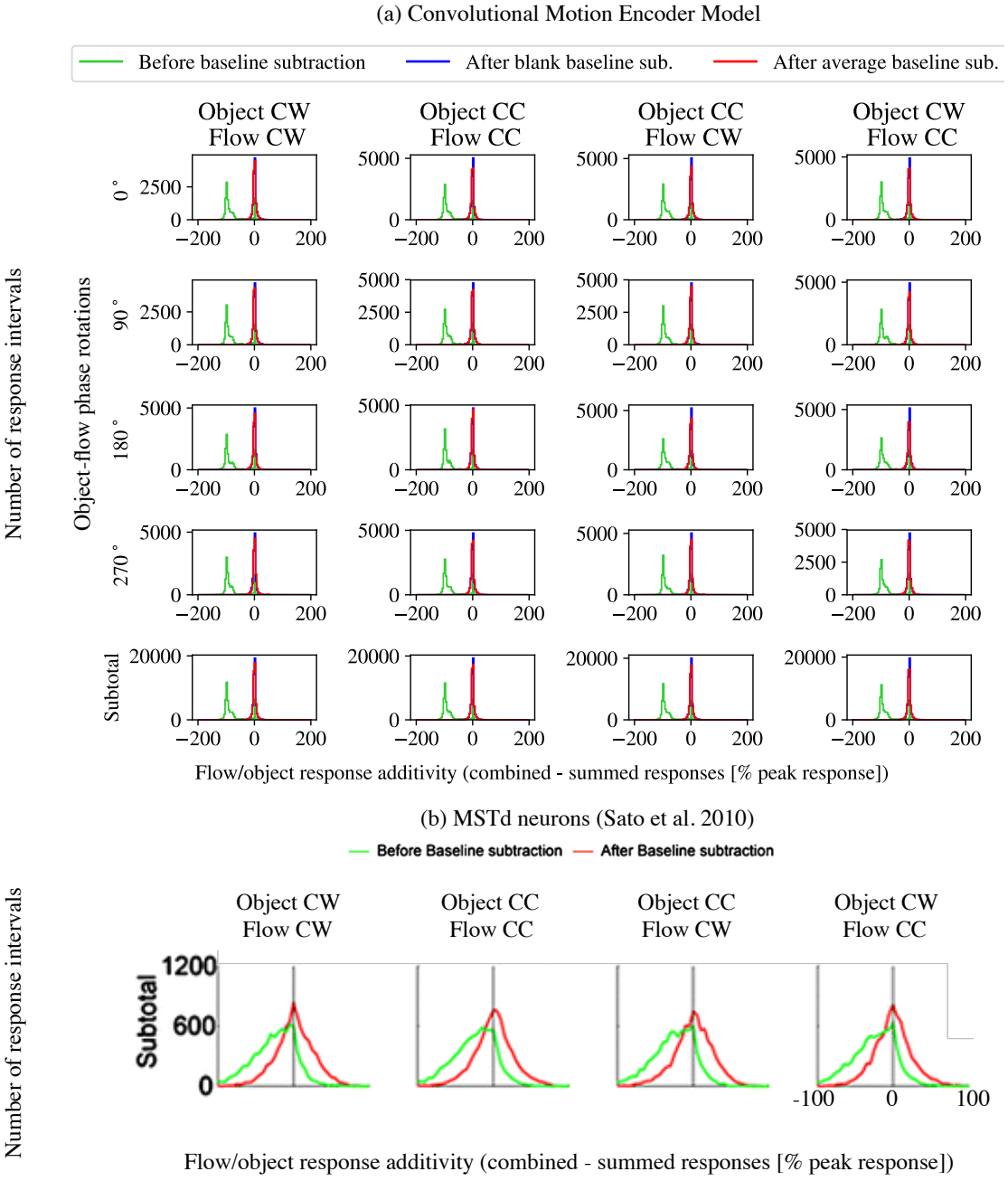


Figure 4.4: Distribution of normalized response additivities of the population of MSTd-like model neurons for the sixteen combined stimulus conditions (a, rows 1 - 4) and the four relative directions accumulated across the four relative phase conditions (a, row 5) and of MSTd neurons (Reprinted from Sato et al. (2010)) for the four relative directions accumulated across the four relative phase conditions (b). In each subplot, response additivity (abscissa) is plotted as the difference between normalized combined and summed responses of all neurons across all response intervals (ordinate) in that stimulus condition.

activations generated by optic flow and object motion presentations was subtracted from the activation generated by the corresponding combined stimulus. This normalized activation differential is termed as response additivity (Sato et al., 2010). We combined responses across all model neurons to test the net effect of combining optic flow and object motion stimuli on the whole population.

Figure 4.4 depicts the distribution of normalized response additivities of the population of MSTd-like model neurons for sixteen combined stimulus conditions (Figure 4.4(a), rows 1 - 4). Each column represents one of the four relative directions between self-movement and object-movement and each row represents phase shift in the combined stimuli. In each subplot, response additivity (abscissa) distribution is plotted across all model neurons, stimuli conditions, and response intervals (ordinate), before and after baseline activity subtraction. Neuron activity interactions for the whole population do not show any distinctive response to any single stimulus condition, implying the absence of any population bias to certain combinations of optic flow and object motion stimuli. To compare between the effects of an earth fixed object in the two same relative direction conditions vs. an independently moving object in the two opposite relative direction conditions, we plot the cumulative distribution of normalized response additivities across all phase shift conditions for each relative direction (Figure 4.4(a), row 5). The two sample t-test of the cumulative distributions for the fixed object conditions (-76.16 ± 37.098) and the independently moving object conditions (-76.75 ± 36.81) reveals that the distributions of response additivities are statistically different at significance level of 0.05. However, the overlapping coefficient (OVL) of the cumulative distributions for the four relative direction conditions is 0.9761, in the scale of 0-1, with OVL=1 for identical normal distributions (Inman and Bradley Jr, 1989). We can conclude that the model population response additivity is sensitive to the relative object motion component in stimulus input, however there is no overall bias toward fixed vs. independently moving objects. For the MSTd neurons recorded by Sato et al. (2010), the OVL of the cumulative response additivity distributions for the same set of stimulus condi-

tions is 0.9611. Therefore, the agreement between the distributions of normalized response additivity for relative object motion variations is suitably captured by our model.

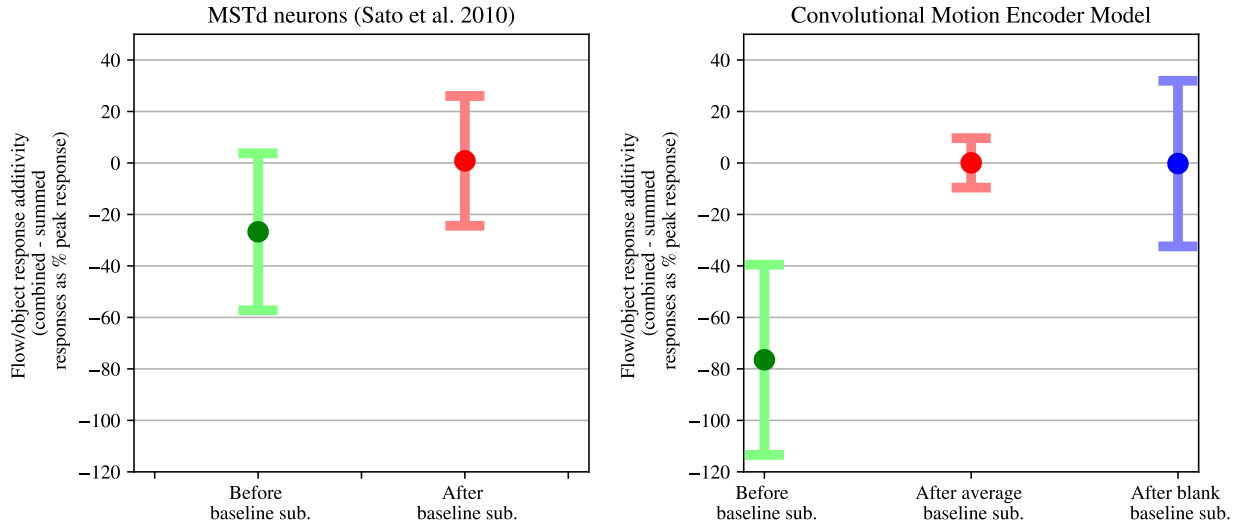


Figure 4.5: Distributions of normalized response additivity of MSTd and model neurons across all sixteen combined stimulus conditions and the corresponding shifted distributions after baseline subtraction. The green bars represent distributions before subtracting baseline neuron activations and the red/blue bars represent distributions after subtracting baseline activations. MSTd response additivity distributions were generated using the mean and standard deviation provided by Sato et al. (2010) across all sixteen combined stimulus conditions.

Similar to Sato et al. (2010), we observed that combining object motion and optic flow results in a sub-additivity of activations, indicated by negative normalized response additivity values. This occurred because the sum of neuron activations to the presentations of object motion and optic flow stimuli alone was larger than the activation for the corresponding combined stimulus. The mean sub-additivity across the sixteen combined stimulus conditions ranged between -75% and -78% of the activations normalized to the peak neuron responses. Figure 4.5 depicts that the MSTd neurons recorded by Sato et al. (2010) also showed a sub-additivity of spike responses for combined stimulus before baseline subtraction. However, it was observed that subtraction of the baseline firing rates from the combined and alone conditions, which is calculated as the average firing rate of the MSTd neurons during a 250 ms response interval prior to stimulus onset without visual motion stimuli, caused a sub-

stantial distribution shift toward super-additivity, indicated by positive normalized response additivity values. A similar effect is also observed for our population of MSTd-like model neurons after subtraction of either average baseline activations or mean baseline activations, shown in Figure 4.5, resulting in a positive shift in mean additivity to the range of -2% to 1%.

In case of MSTd neurons, baseline subtraction also reduces the standard deviation from 30.5% to 25.2%, thus narrowing the distribution of normalized response additivity. For our MSTd-like model neurons, the standard deviation of normalized response additivity distribution is 37.0% before baseline subtraction, 9.6% after average baseline subtraction, and 32.2% after blank baseline subtraction. Although both types of baseline subtraction narrows the distribution, the effect is smaller for blank baseline subtraction. However, the range between 5th and 95th percentile of the distribution is similar after both baseline subtractions, as shown in Figure 4.5.

4.3.2 Population Response Interactions with Combined Stimuli

We perform multiple linear regression analysis to analyze the neuron activity interactions, by fitting neuron activations during the sixteen combined stimuli conditions as a function of the activations to their corresponding object motion and optic flow stimuli conditions. We first fit a multiple regression model with these two factors for each of the 318 MSTd-like neurons selected in the methods section based on their significance to response interactions from the whole population of the conv5 layer neurons. At first, the factors are taken as is, without any baseline subtraction and without any constant term. The distribution of regression fits is depicted in Figure 4.6 (b), with mean $R^2 = 0.44$. Although, all the neurons produce significant fits with p-values < 0.05, only 44% neurons yielded $R^2 > 0.3$. The distribution is clearly bimodal with a group of neurons with R^2 value close to 1 and the other group with

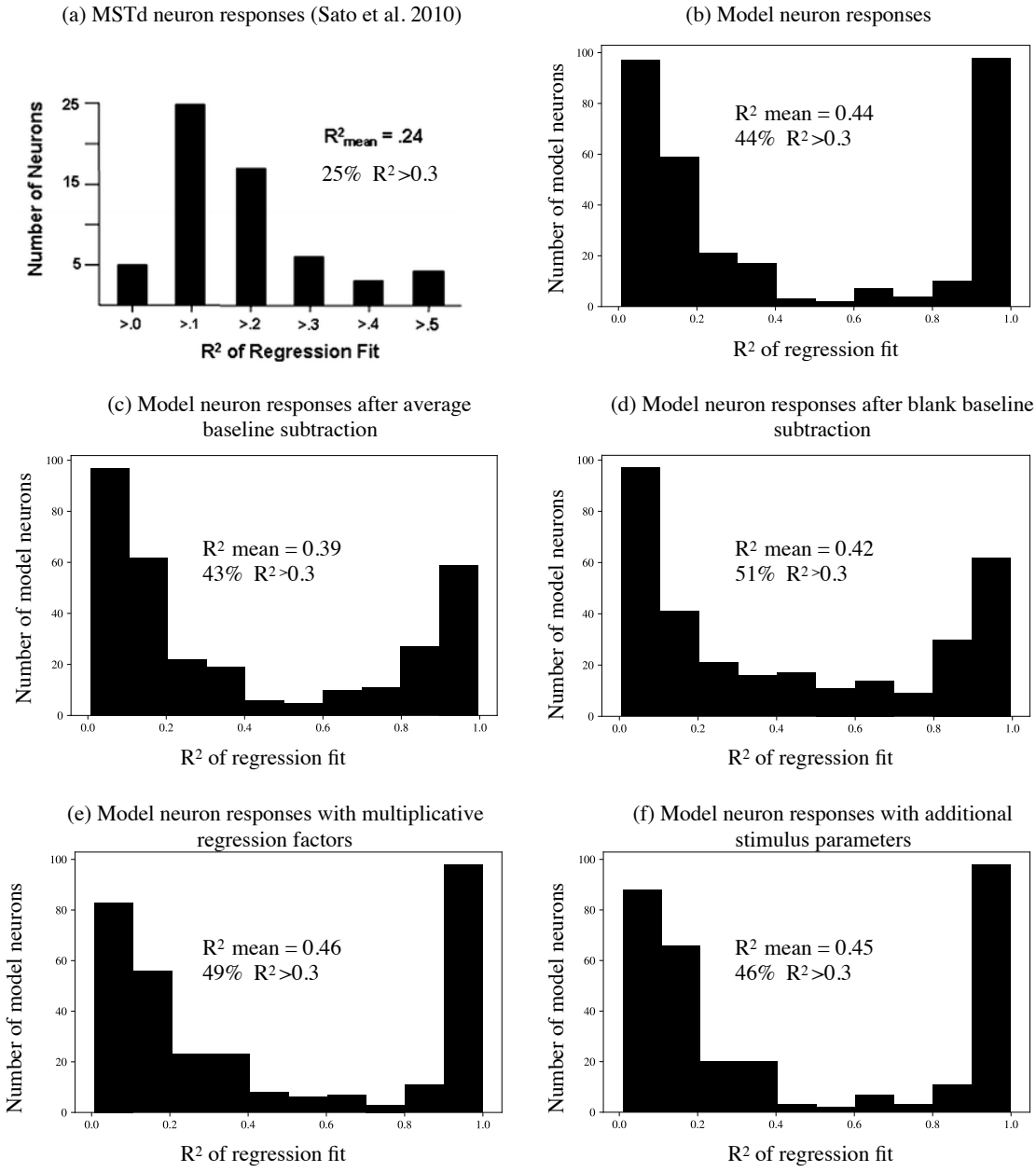


Figure 4.6: The distribution of regression fits of responses to combined stimuli in terms of responses to optic flow and object motion stimuli, their multiplicative interactions, and other stimulus parameters. (a) The distribution of fits produced by MSTd neuron responses (reprinted from (Sato et al., 2010)). Other frames depict the distribution fits based on model neuron responses (b), after average baseline subtraction (c), after blank baseline subtraction (d), with multiplicative factors (e), and with additional stimulus parameters (f).

a wider mode toward small R^2 values.

The distribution of fits obtained from MSTd neurons, considering higher order polynomial terms, such as $flow^2$ and $flow \times object$, is depicted in Figure 4.6(a). The MSTd neurons have a mode near $R^2 = .1$, however there is not a group with large R^2 values. To test the effects of using higher order polynomial terms on the distribution of variance explained by activations to alone stimuli, we added the factors $flow^2$, $object^2$, and $flow \times object$ to our regression. Figure 4.6 (e) shows that adding these extra higher order terms only slightly increases the overall R^2, however 5% additional neurons yielded a fit of $R^2 > 0.3$. This is similar to the finding in (Sato et al., 2010) that these higher order terms produce better fits for some individual neurons, however, there is only modest general benefit on the overall quality of fit across all neurons and conditions. We also tested additional stimulus parameters, viz., object location, relative direction between object and flow, and phase rotation between object and flow, on their effect on regression fits. We found that the addition of these additional stimulus parameters has only modest effect on the fits ($R^2 = 0.45$). The p-values of these stimulus parameters indicate their relative significance toward responses to the combined stimuli. For 138/318 neurons, the relative direction between object and flow yields p-values < 0.05, whereas the phase rotation between object and flow and object location parameters are equally significant for fits of only 66 and 73 neurons, respectively.

We analyzed the effect of baseline subtraction from model neuron activations on the quality of regression fits. Figure 4.6(c) shows that when the average activation across all alone and combined stimulus conditions is subtracted for each neuron, it neither helps in increasing the fits of individual neuron activations to the combined stimuli (43% neurons yield $R^2 > 0.3$) nor improves the net explained variance by the population ($R^2 = 0.39$). However, subtraction of the average neuron activity without any presentation of visual motion stimuli, which we call as blank baseline subtraction, improves the fits of individual neuron activations to the combined stimuli as 51% neurons produce a fit with $R^2 > 0.3$. This baseline subtraction also reduces the number of neurons with R^2 close to 1, and distributes the variance explained across a larger population of neurons.

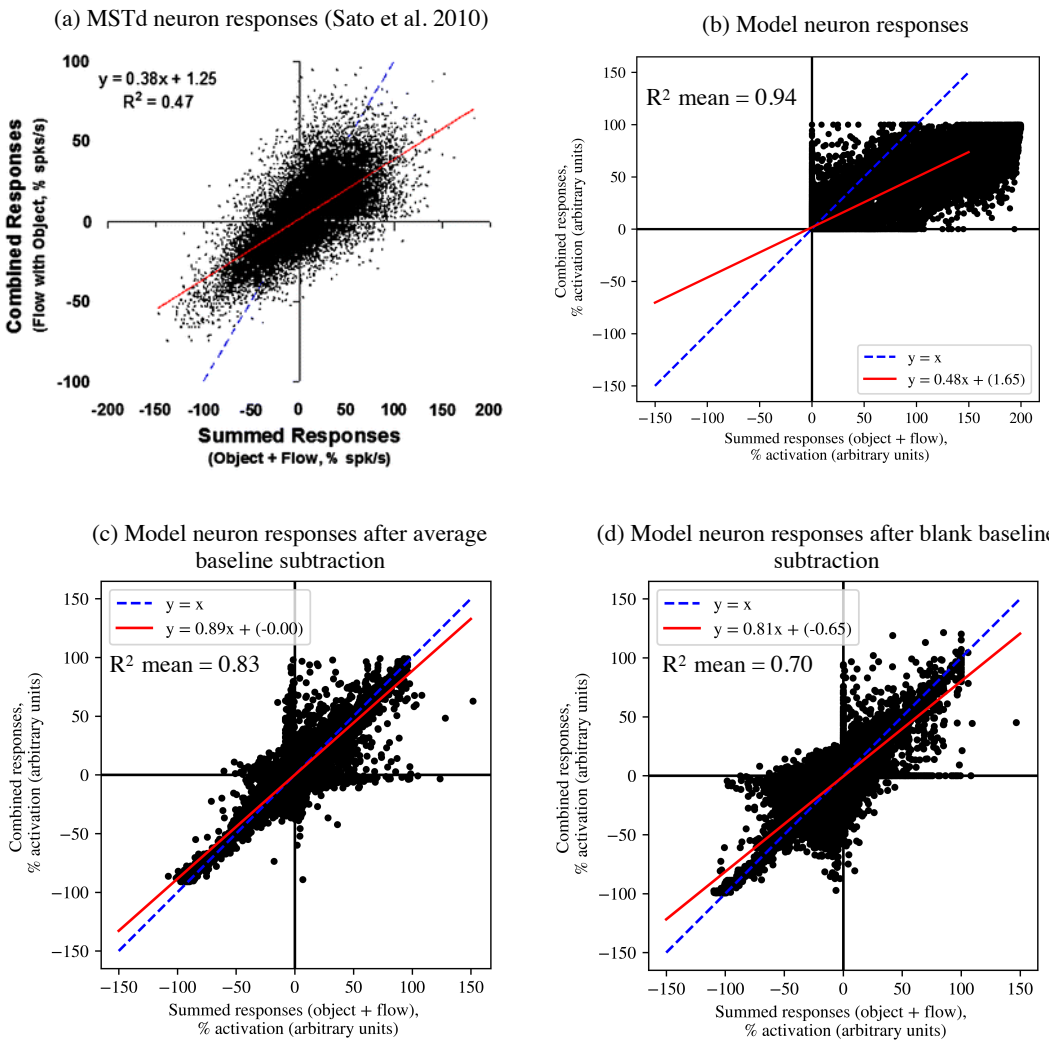


Figure 4.7: Population response additivity of MSTd neurons (a, reprinted from (Sato et al., 2010) and of MSTd-like model neurons before (b) and after (c, d) baseline subtraction. In each panel, the blue dashed line denotes the additivity line, the region above it represents super-additivity and the region below it represents sub-additivity. The solid red line is the regression fit for combined responses in terms of the summer object-only and flow-only responses. Responses are normalized to the largest response elicited by the neuron for any of the object-only, flow-only, or combined stimuli.

It is possible that the response additivity variance of our model is not explained completely by single neurons and instead represented at a population level. To test this hypothesis, we plotted the response to the combined stimulus against the sum of responses to corresponding object-only and flow-only stimuli, for each response interval for each stimulus condition across the full neuron population in Figure 4.7. It can be seen that response additivity has

a better fit at the population level, given by larger R^2 values compared to individual neuron fits in Figure 4.6, for both MSTd neurons (Sato et al., 2010) and our MSTd-like model neurons. The model neuron responses are non-negative due to ReLU non-linearity and indicate a sub-additive interaction between combined and summed responses as most data points lie below the additivity line (Figure 4.7(b)). However, before baseline subtraction, we cannot characterize the differences between excitatory and inhibitory responses, in which stimulus presentation increases and decreases the response magnitude compared to the baseline responses, respectively. For both average baseline subtraction and blank baseline subtraction, depicted in Figure 4.7(c) and (d) respectively, the excitatory responses yield more sub-additive interactions and the inhibitory responses yield more super-additive interactions, evident from the slope of less than 1 of the regression lines. The effect is more prominent for blank baseline subtraction and is also observed for MSTd neurons (Figure 4.7(a)). Some interval responses of the model neurons also indicate additive interactions, particularly for inhibitory responses. Another observation from the model neuron responses after baseline subtraction is that some responses lie along the x and y axes, indicating selectivity for either combined or individual stimuli, but not for both. The small intercept in each regression fit characterizes the additivity of near baseline activity observed for most neurons during stimulus presentation as not having influence on the responses to combined stimuli, similar to the MSTd neurons (Sato et al., 2010).

4.3.3 Stimulus Specific Selectivity

To visualize stimulus specific tuning properties of individual model neurons, we plot their normalized activations during stimulus presentations of the two object motion conditions, the two optic flow conditions, and the sixteen combined stimulus conditions.

Many neurons in the model responded to a combination of self-motion and object motion

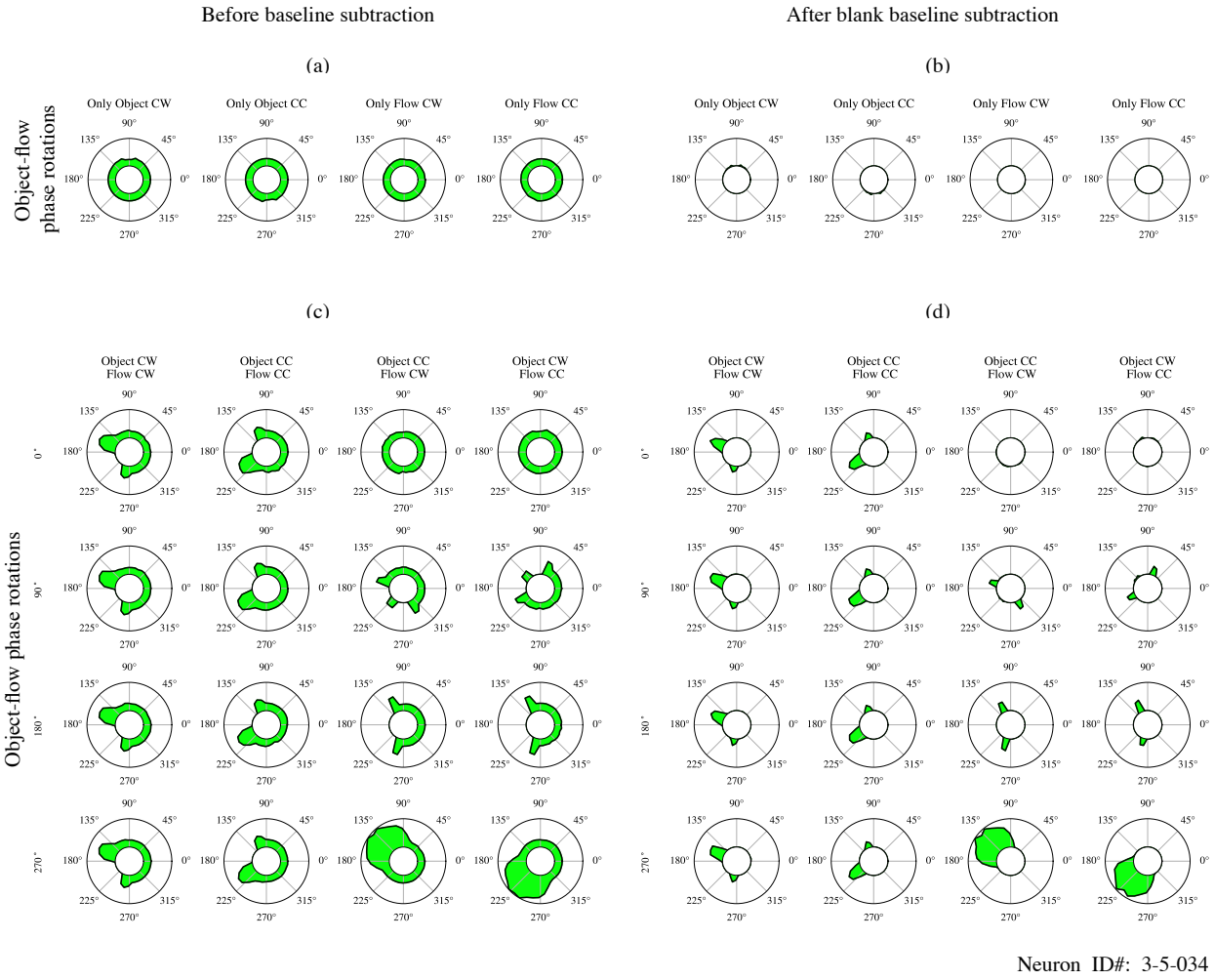
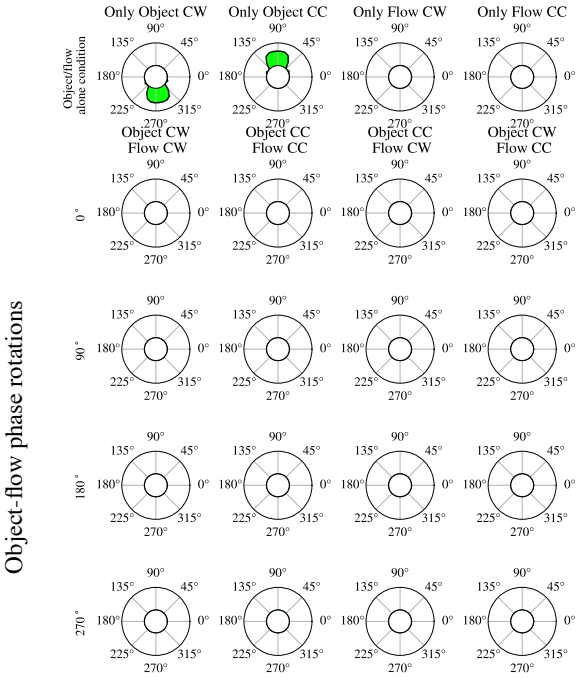


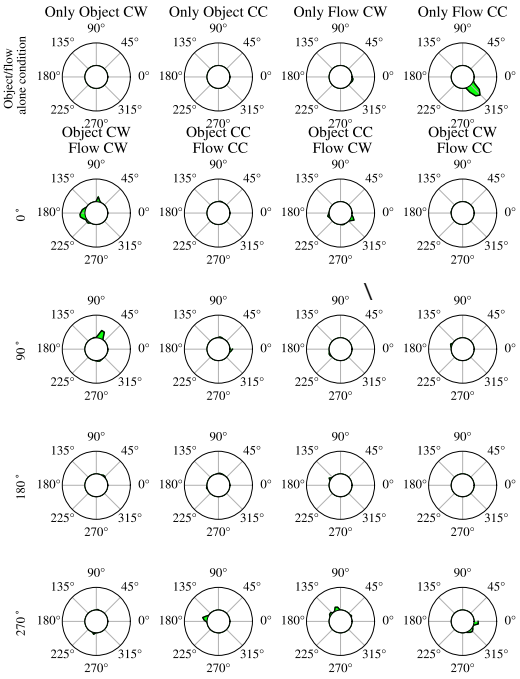
Figure 4.8: MSTd-like neuron that responds to combined object and self-motion stimuli. Circular activation plots of responses by a MSTd-like model neuron to 20 combined and alone stimulus conditions. Location around each circle corresponds to the position of the observer on a circular trajectory of self-movement when the activation was recorded. All activations are normalized to the maximum activation of the neuron across all stimulus conditions. (a) activations for object motion and optic flow stimuli, (b) activations for object motion and optic flow stimuli after blank baseline subtraction, (c) activations for sixteen combined stimulus conditions, (d) activations for sixteen combined stimulus conditions after blank baseline subtraction.

cues. Figure 4.8 shows a representive response. The plot depicts the activations of a single MSTd-like model neuron in response to the 20 stimulus conditions, before and after baseline subtraction. (Figure 4.8 (a) and (c)), the neuron produces at least some base levels of activation at most locations along the circular track in all 20 stimulus conditions, and

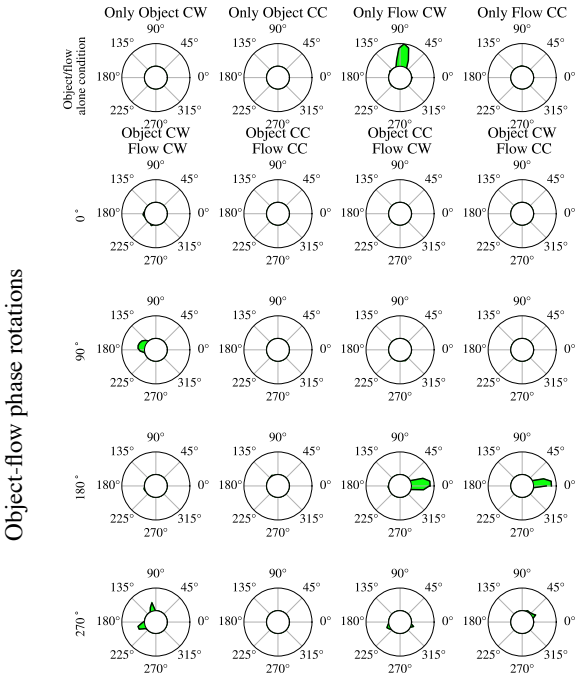
(a) Neuron with object motion preference (ID#: 2-4-052)



(b) Neuron with ego-motion preference (ID#: 1-3-098)



(c) Neuron with dual preference (ID#: 2-2-230)



(d) Neuron without specific preference (ID#: 6-6-158)

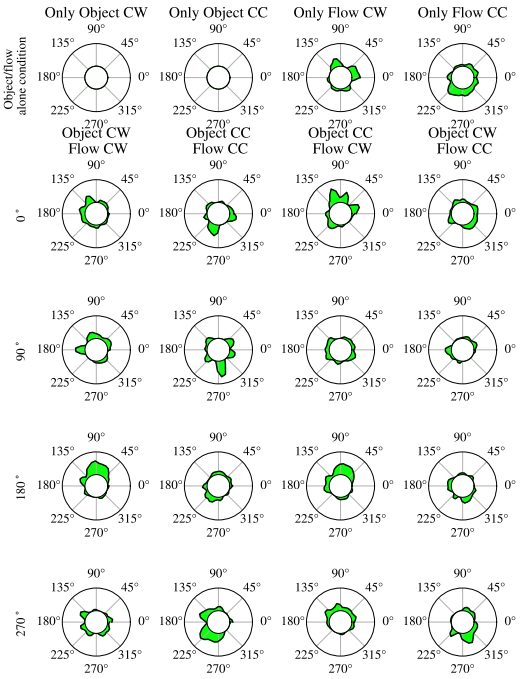


Figure 4.9: Circular activation plots of activations of four MSTd-like model neurons. Each neuron prefers either object motion (a), ego-motion (b), or both ego-motion and combined stimuli (c), or has no preference (d). Activations are shown after blank baseline subtraction.

distinctively larger activations at around 135° of the circular trajectory in the Object CC - Flow CW condition and at around 225° of the circular trajectory in the Object CW - Flow CC condition. After blank baseline subtraction, this distinct selectivity becomes prominent (Figure 4.8 (b) and (d)). The neuron produces a preferred response for a specific combination of object motion and optic flow, but little response for the other stimulus conditions. This is consistent with the observation by Sato et al. (2010) that most MSTd neurons produce much larger response in one or two of the combined stimulus conditions compared to the alone and other combined stimulus conditions. We do not show the circular activation plots of the neuron after average baseline subtraction, as they look visually similar to the activation plots before baseline subtraction due to normalization.

Figure 4.9 depicts activations of four representative MSTd-like model neurons, each with preferences for different types of stimulus conditions. Figure 4.9(a) shows a neuronal response specific to object motion. Figure 4.9(b) shows a neuronal response specific to self-motion. Figure 4.9(c) shows responses to a specific combinations of object and self-motion. Some neurons do not show a strong preference for a specific stimulus condition and are activated by many types of stimuli (Figure 4.9(d)).

Given the distinctive activation profile of neurons like that shown in Figure 4.8, we would like to know the distribution of preferred object motion and optic flow in different stimulus conditions across the neuron population. Figure 4.10 depicts the selectivity of individual model neurons in all combined and alone stimulus conditions after baseline subtraction. A small subgroup of the neuron population responds to variations of object motion stimulus (Figure 4.10 (a), left two circular plots). However, mostly a different neuron subgroup responds to the range of optic flow stimuli (Figure 4.10 (a), right two circular plots). This can be seen based on green/blue colored neurons being more responsive to object motion stimuli and violet/red colored neurons being more responsive to optic flow stimuli. The coherence in color (i.e. neuron order) may arise from the spatial organization of the convolutional

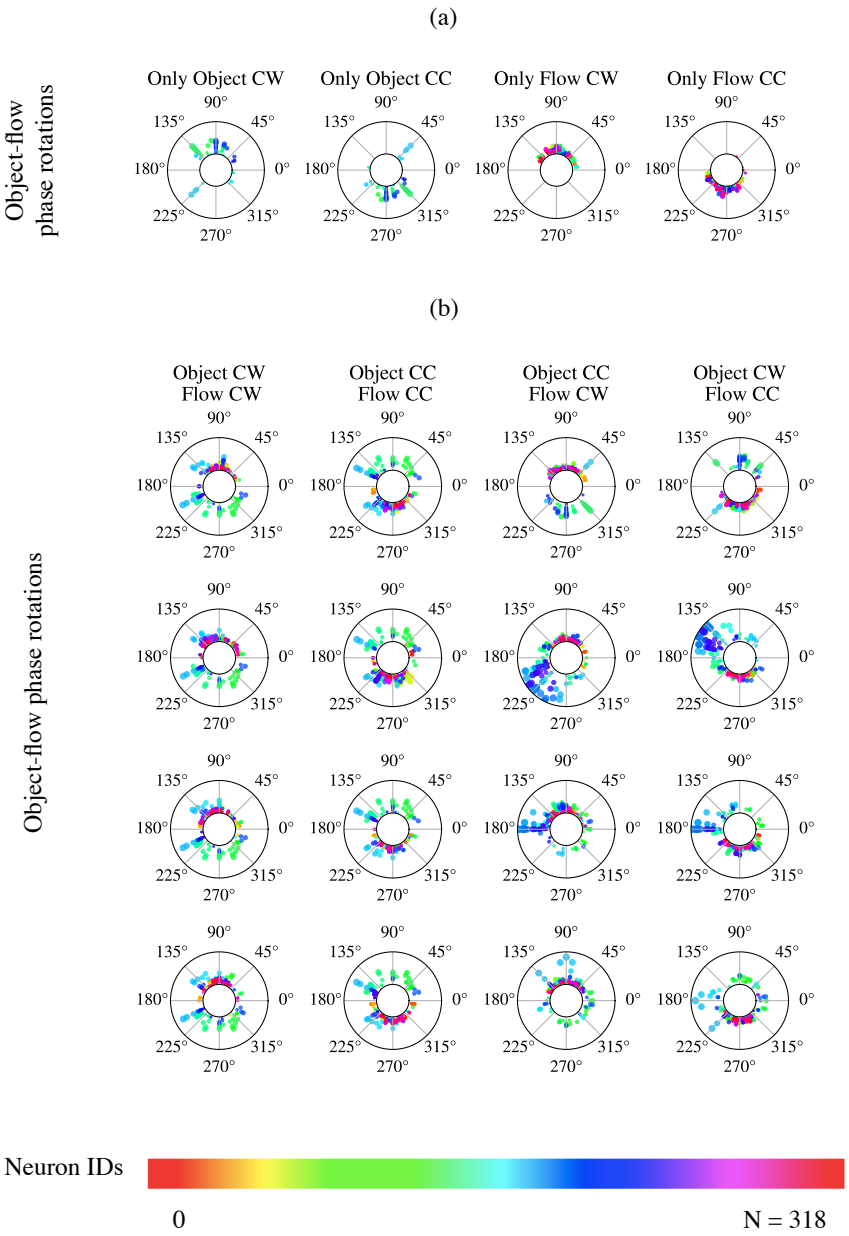


Figure 4.10: Stimulus specific selectivity of all neurons to 20 combined and alone stimulus conditions after blank baseline subtraction. Location around each circle corresponds to the position of the observer on a circular trajectory of self-movement simulated using visual stimuli. Each filled circle in each plot represents one neuron, its angular location defines the preferred motion stimulus corresponding to that location, its radial distance defines the normalized activation magnitude to the preferred stimuli. Area of each circle is proportional to its radial distance. Each neuron is color tagged from the colorbar shown below and is consistent across all circular plots. (a) Selectivity for object motion and optic flow stimulus conditions. (b) Selectivity for the sixteen combined stimulus conditions.

kernels. For both groups, the preference distributions are non-uniform. Also, the preference distributions have almost 180° phase shift between the clockwise and anticlockwise stimulus directions in both object motion and optic flow conditions. The preference distributions are more diverse and dense for combined stimuli and vary for different phase rotations between object motion and optic flow. A wide distribution of selectivity can be seen for the combined stimulus conditions with optic flow and object motion (Figure 4.10 (b)).

Table 4.1: Categorization of MSTd-like model neurons as having preference to one of the three stimulus categories: object motion, optic flow, and combined stimuli.

Threshold (T_c)	Ratio (R_c)	Baseline subtraction	# neurons with object preference	# neurons with optic flow preference	# neurons with combined stimuli preference	Total (/318)
0.05	1	None/Avg. baseline	35	24	259	318
		Blank baseline	9	14	263	286
	2	None/Avg. baseline	0	0	26	26
		Blank baseline	3	3	174	180
0.1	1	None/Avg. baseline	35	24	259	318
		Blank baseline	8	11	254	273
	2	None/Avg. baseline	0	0	26	26
		Blank baseline	2	1	166	169

We divided the MSTd-like neuron population into three categories as preferring either object motion, optic flow, or combined stimulus conditions, using the categorization protocol given in Equation 4.8. We varied the threshold parameters T_c and R_c and baseline subtraction method to see their effect on the number of neurons that can be categorized into one of the three categories, the results are shown in Table 4.1. The threshold parameter $T_c = (0.05, 0.1)$ controls if a neuron can be classified into a category based on its maximum activation to all stimuli in that category. Therefore, a higher $T_c = 0.1$ results in fewer (or equal) number of neurons categorized than $T_c = 0.05$. Arguably, the R_c parameter is more interesting since it weighs the relative activations among categories for decision making. $R_c = 1$ implies that a neuron passing the threshold criterion will be assigned to the stimulus category that activates it maximally. However, it does not guarantee a strong preference to that category

over other categories and the neuron could be almost equally responsive to more than one categories. On the other hand, $R_c = 2$ guarantees that a neuron is at least twice as active for the preferred category than all other categories. Therefore, fewer neurons get categorized and all of them have a strong preference for only one category. In our analysis, we assume $R_c = 2$ randomly to know how many neurons have a preference for one type of stimuli, however this can be adjusted to other numbers greater than 1.

For $T_c = 0.1$ and $R_c = 1$, all MSTd-like model neurons are assigned a maximal activation category before baseline subtraction and after average baseline subtraction. As R_c is updated to 2, only 26 out of 318 neurons can be assigned as having a strong preference to a stimulus category. In contrast, after blank baseline subtraction, the same set of parameters yield 273 neurons categorized as having preference for one or more category and 169 neurons categorized as having a strong preference for one stimulus category. However, the number of neurons preferring alone stimulus conditions is small compared to number of neurons preferring combined stimulus conditions, 3:166 for blank baseline subtraction and 0:26 for average or no baseline subtraction.

Taken together, these results suggest that motion can be accurately predicted from populations of neurons that show mixed selectivity to a range of visual stimuli.

4.4 Discussion

Perception of dynamic scenes requires parsing the motion cues due to observer and object movements. Our results suggest that like the dorsal visual stream of the primate brain, an efficient means of encoding these features is through neurons that respond to combinations of motion cues. Specifically, we found that a deep CNN (LeCun et al., 1998; Krizhevsky et al., 2012) trained over a range of object motion and ego-motion stimuli can accurately predict

the 6DoF ego-motion parameters and the object motion. The CNNs representations emerge from early visual processes that calculate motion from image sequences (e.g. in LGN and V1) and extract depth from binocular disparity (e.g. in V1 and V2). These representations account for a range of behaviors exhibited by MSTd neurons in response to optic flow, object motion, and combined stimuli, such as: (i) sub-additivity of response interactions between alone and combined stimulus conditions, (ii) shift towards super-additivity and narrower interaction distribution after baseline subtraction, (iii) unbiased population response interaction across all stimulus conditions, (iv) absence of multiplicative interactions across the neuron population, and (v) larger activation by combined stimuli than those evoked by component stimuli for most neurons (Sato et al., 2010). Similar to the MSTd neurons, the model neurons are selective to a subset of the stimuli, preferring certain stimulus intervals of certain combined or alone stimulus conditions. However, at the population level, there was no significantly large response toward any of the combined stimuli conditions, including the congruent CW-CW and CC-CC conditions with phase aligned object motion and optic flow.

Recently, variants of goal driven deep neural network models have revolutionized computational neuroscience with their ability to predict neural responses of higher order cortical areas, explaining various response characteristics observed in both humans and monkeys (Yamins et al., 2014; Cadieu et al., 2014; Khaligh-Razavi and Kriegeskorte, 2014; Güçlü and van Gerven, 2015; Cichy et al., 2017). However, these models have so far been investigated for predicting ventral visual pathway responses to image input and not for the dorsal visual pathway that processes motion between sequence of images. As such, these models are not tested to explain neural responses to multi-component inputs, such as combined optic flow and object motion stimulus considered here, and the interactions between responses to individual components.

Using multi-component motion stimuli, Sato et al. (2010) found that after baseline extraction the interactions between responses to individual stimulus components were revealed. Simi-

larly in our model, baseline subtraction fits the sub-additive population response interaction model and highlighted the preference of individual neurons to certain stimulus categories. Sato et al. (2010) performed a similar baseline subtraction procedure based on average neural activity in absence of visual stimuli. Although baseline activations without visual input in our model is different than spontaneous MSTd activity when no visual stimuli is present, their removal lead to better match of response characteristics to the same set of stimuli, such as shift toward interaction additivity and narrower interaction distribution around zero, as well as better individual neuron level specificity.

Early models of cortical visual motion processing incorporated an opponent stage of processing, i.e. motion detectors tuned to opposite direction of motion inhibit each other (Van Santen and Sperling, 1985; Adelson and Bergen, 1985). This has been extended by other models as a mechanism to recover object velocity from optic flow (Layton and Fajen, 2020; Warren and Rushton, 2009). However, several psychophysical studies have shown that perception of moving patterns is unaffected by addition of motion in the opposite direction (Levinson and Sekuler, 1975; Watson et al., 1980; Dobkins and Teller, 1996; Raymond and Braddick, 1996), suggesting antagonistic motion suppression may not be employed by higher order visual cortical areas (Krekelberg and Albright, 2005). Thiele et al. (2000) found that for a majority of MT neurons, responses to preferred motion are not affected by simultaneous presentation of motion in the anti-preferred direction. The neurophysiology study by Sato et al. (2010), which used the same visual stimuli as ours, did not observe any suppression effect for opposing optic flow and object motion conditions or any facilitation effect for congruent optic flow and object motion conditions in the population of MSTd neurons. In agreement with that finding, our model neurons did not exhibit any suppression or facilitation effects for opposite and congruent stimuli. An alternative theory states that the direction of motion in presence of additional opposing motion is perceived as weighted average activity of independent motion analyzers without lateral inhibitions (Raymond and Braddick, 1996). Our model extends this theory to perception of both motion direction and speed by a hierarchy of

independently activated convolutional model neurons without lateral or top-down inhibition, whose activations are decoded linearly to estimate ego-motion and non-linearly to estimate object motion.

Our model provides a computational account of a wide range of response characteristics observed in MSTd neurons when presented with diverse combinations of optic flow and object motion stimuli (Sato et al., 2010). These behaviors emerged automatically from a generic goal driven optimization for the tasks of object and ego-motion prediction without any prior assumptions about model behavior. In future studies, the model could be examined further for more naturalistic stimuli and motion conditions (Geiger et al., 2012; Wulff et al., 2017; Lv et al., 2018) and efficiency of representation (Olshausen and Field, 1997). Additionally, the significance of the components in the proposed convolutional encoder and the loss terms in Equation 4.1 could be studied by comparing to standard CNN blocks trained on large image datasets (Simonyan and Zisserman, 2014; He et al., 2016).

Chapter 5

Recurrent Neural Network Model of Pursuit Eye Movement for Visual Tracking

5.1 Introduction

Cortical motion perception, which was modeled in the previous chapter, provides an input signal to an important sensorimotor system in the brain, which is the eye movement control system for tracking rapidly moving targets. Primates are incredibly proficient at tracking a visual target with their eyes. Since their foveal vision is narrow, they rotate their eyes continuously in the direction of a moving object to keep it centered on the fovea, which provides high acuity information (Varjú and Schnitzler, 2012). This type of eye movement is known as smooth pursuit, as eye velocity changes smoothly in response to target velocity. The pursuit system is able to track with almost zero lag between eye and non-linear target velocities (Barnes et al., 1987). This behavior is particularly impressive, since due to sensory

and processing delays of 80-100 ms in the visual pathways (Krauzlis and Lisberger, 1994). A system that solely relies on retinal error feedback for oculomotor movements cannot achieve zero lag pursuit. Furthermore, the primate eyes continue smooth pursuit of a target after its disappearance (Eckmiller and Mackeben, 1978; Whittaker and Eaholtz, 1982). These two pursuit behaviors indicate a predictive mechanism that is able to generate current and future eye velocities based on the target motion sequence in the past (Barnes et al., 1987).

Early control-theoretic models of smooth pursuit by Robinson et al. (1986); Krauzlis and Lisberger (1994) did not consider the predictive capabilities of pursuit, rather they tried to mimic the experimentally observed typical initial acceleration, overshoot, and response latency, while tracking a simple constant velocity stimulus. Later models used prior knowledge or memory from previous trials to eliminate sensory delays and to be able to continue pursuit during occlusions (de Xivry et al., 2013; Deno et al., 1995; Barnes and Wells, 1999). However, these memory-based models are not biologically plausible since they rely on periodicity of the target motion, resulting in periodic improvement of pursuit lag Shibata et al. (2005). They also cannot adapt to transient perturbations in target velocity immediately, whereas humans adapt to perturbation and phase shift of a sinusoidal target within a cycle (Van den Berg, 1988).

A fundamental problem associated with pursuit is to generate eye velocity that persists while target velocity on retina is zero. This takes place during perfect zero lag tracking and during target occlusion. The neural mechanism that generates predictive eye velocity signals in absence of visual inputs is not known from the existing pursuit models. The most relevant Kalman filter based predictive pursuit model by de Xivry et al. (2013) cannot generate persistent eye velocity during long occlusions of a target with nonlinear velocity, due to absence of the error feedback to correct filter predictions. In contrast, both humans and monkeys were found to continue pursuit of a target with sinusoidal velocity after disappearance (Eckmiller and Mackeben, 1978; Whittaker and Eaholtz, 1982).

In this chapter, we propose a recurrent neural network (RNN) model of predictive smooth pursuit eye movement that rapidly learns the target velocity sequence and generates self-sustained predictive eye velocity signals to track a moving target with near zero lag. The model is able to i) gradually eliminate the initial lag between eye and target velocities, ii) track an occluded target with nonlinear velocity, iii) adapt to unpredictable perturbation and phase shift in target velocity, and iv) qualitatively reproduce the typical initial pursuit acceleration observed in primate smooth pursuit experiments (Van den Berg, 1988; Whittaker and Eaholtz, 1982; Eckmiller and Mackeben, 1978; Keating, 1991).

This chapter is based on previously published work:

Kashyap, H. J., Detorakis, G., Dutt, N., Krichmar, J. L., & Neftci, E. (2018). "A Recurrent Neural Network Based Model of Predictive Smooth Pursuit Eye Movement in Primates". In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN) (pp. 5353-5360).

Portions are reprinted with permission, © 2018 IEEE.

5.2 Methods

Studies with human and non-human primates have shown that the predictive pursuit system learns the spatio-temporal sequence of target velocity (Whittaker and Eaholtz, 1982; Eckmiller and Mackeben, 1978; Van den Berg, 1988). For this, the predictive system in the brain solves three challenges that arise from biology and the nature of the visual tracking task. First, the target velocity sequence is learned rapidly (e.g. within a few cycles for periodic target velocity (Barnes, 2008)) to eliminate pursuit lag. Second, the learning occurs using Retinal Slip (RS) information that is delayed by 80-100 ms during sensory processing. Third, RS vanishes during zero lag pursuit or occlusion, and therefore eye velocity is

generated internally in absence of external visual cue of target velocity.

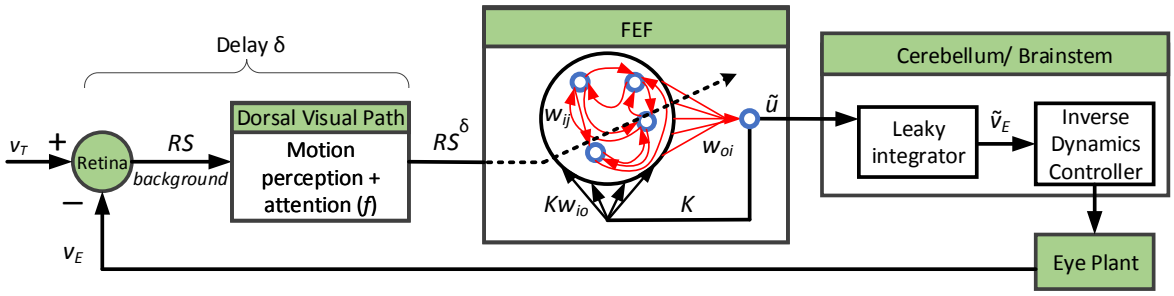


Figure 5.1: The proposed model for predictive smooth pursuit eye movement generation in primates. The plausible brain regions performing the specific functions in the pursuit pathway are shown in green colored boxes. The retinotopic RS is extracted from visual field with a time delay of δ by the dorsal visual pathway (RS^{δ}). A recurrent network of neurons (blue circles) in the FEF region uses RS^{δ} to learn the target velocity sequence and generates $\tilde{u}$, which is then low pass filtered by a leaky integrator to obtain eye velocity predictions ($\tilde{v}_E$). All red colored synaptic connections are modified during learning. Cerebellum and Brainstem together implement an inverse dynamic controller to generate the final eye velocity (v_E) via occulomotor control.

Considering these challenges, we propose a smooth pursuit model using an RNN to rapidly learn the target velocity sequence and predict eye velocity during both the presence and absence of retinal inputs. The model produces eye velocity prediction from spontaneous neural activations and uses a delayed RS as the error signal for learning. The eye velocity prediction is then converted to actual eye motor movements by a downstream Inverse Dynamics Controller (IDC). Figure 5.1 depicts the complete predictive smooth pursuit model.

5.2.1 Neuron Model and Network Architecture

The target velocity on the retina, i.e. RS, results from actual movement of the target in the three-dimensional world and/or due to eye, head, or body movements. Since target motion projected onto the retina is relative to eye motion, the resulting RS is the difference between head-centered target velocity (v_T) and eye velocity (v_E). When the eye perfectly tracks a

moving target with foveal vision, RS will be zero.

$$RS(t) = v_T(t) - v_E(t) \tag{5.1}$$

The retinal output, where RS of the target is embedded in a visual scene background, is transmitted through two interconnected visual cortical pathways, one that recognizes objects (ventral pathway) and the other that extracts motion components in the visual field (dorsal pathway). In Figure 5.1, we only show the dorsal pathway as it processes motion information. The dorsal visual pathway extracts motion of all objects in visual field and the attention system selects the target motion component from background, in time δ. We term the cumulative operations performed by the dorsal pathway as f and its target velocity output as RS^{δ}.

$$RS^{\delta}(t) = f(RS(t - \delta)) \tag{5.2}$$

where, RS^{δ} is delayed by time δ since its projection on the retina.

The RNN, shown inside the box labeled FEF in Figure 5.1, uses RS^{δ} as the error signal to learn target velocity sequence online and generates predictive eye velocity signals. Similar predictive activities during pursuit have been observed in the FEF region of frontal cortex, which receives outputs from the dorsal pathway (Fukushima et al., 2002; Keating, 1991; MacAvoy et al., 1991). RS^{δ} vanishes as the lag between target and eye velocities is reduced during learning, therefore the RNN needs to predict eye velocity in absence of visual inputs.

The implemented RNN is a type of reservoir computing (Jaeger, 2001; Maass et al., 2002). The RNN has 500 neurons connected all-to-all and operates in a chaotic regime. All neurons of the RNN connect to a single readout neuron (o) via readout synapses. The recurrent and readout synapses are plastic and are modified online using RS^{δ} as the error. The output of

the readout neuron $\tilde{u}$ is fed back to the RNN through fixed random weights with a gain K. The signal $\widetilde{u}$ is low pass filtered to obtain the eye velocity prediction $\tilde{v}_E$.

The neurons in the RNN follow the dynamics proposed by Sussillo and Abbott (2009). The dynamics of neuron i can be written as:

$$\tau \frac{\mathrm{d}x_i}{\mathrm{d}t} = -x_i + \sum_{j \in Pre(i)} w_{ij} r_j + K w_{io} \tilde{u} \tag{5.3}$$

Where, τ is the time constant, x_i is the neuron state, $Pre(i)$ is the set of neurons that projects to post-synaptic neuron i, w_{ij} is the weight of the synapse from neuron j to neuron i, w_{io} is the weight of the synapse from the readout unit to neuron i, and r is the non-linear activation function given as,

$$r_i = \tanh(x_i) \tag{5.4}$$

The readout unit linearly combines neuron activations weighted by w_{oi}, which can be considered as an integrator with unit gain and time constant.

$$\tilde{u} = \sum_i w_{oi} r_i \tag{5.5}$$

The leaky integrator performs low pass filter on the output of the readout unit with gain K_l and time constant τ_l to generate pursuit eye velocity prediction $\tilde{v}_E$, following the dynamics given by:

$$\tau_l \frac{\mathrm{d}\tilde{v}_E}{\mathrm{d}t} = -\tilde{v}_E + K_l \tilde{u} \tag{5.6}$$

5.2.2 Online Learning

The recurrent and the readout weights are updated periodically after every δ time units, same as the visual processing delay. This is done because the effect of weight updates on RS is obtained after δ. The FORCE learning procedure by Sussillo and Abbott (2009) is applied to learn all recurrent and readout weights using the same delayed error signal RS^{δ}. Within the RNN, the recurrent weight update for the synapse from neuron j to neuron i is defined as,

$$w_{ij}(t) = w_{ij}(t-\delta) - RS^{\delta}(t) \sum_{k \in Pre(i)} P_{jk}(t) r_k(t) \tag{5.7}$$

Similarly, the weights from the RNN to to the readout unit are updated as,

$$w_{oi}(t) = w_{oi}(t-\delta) - RS^{\delta}(t) \sum_{k \in Pre(o)} P_{ik}(t) r_k(t) \tag{5.8}$$

where, P is a matrix containing individual learning rates for all synapses, updated regularly. It is initialized to I/α, where α is some constant and I is the identity matrix. P is updated as the inverse of the correlation matrix of neuron activations plus a regularization term αI (Sussillo and Abbott, 2009).

Biological interpretation of this type of neural dynamics has been suggested previously based on experimental data (Sussillo et al., 2007). Cortical neural networks maintain a spontaneous baseline activity, which is chaotic and inherently unstable. However, short-term plasticity based on pre-synaptic firing dynamically tunes these networks to be stable and respond

reliably to external stimuli. This self tuning principle allows these cortical networks to respond to external perturbations with characteristic transient response.

We follow the findings and the theory that the cerebellum and the brainstem together implement an IDC, which cancels the dynamics of the eye plant (Shidara et al., 1993). As in (Shibata et al., 2005), we assume that the IDC is ideal, and therefore we can write,

$$v_E = \tilde{v}_E \tag{5.9}$$

where, $\tilde{v}_E$ is the low pass filtered eye velocity prediction.

5.3 Experimental Results

We test the proposed predictive pursuit model on three characteristic pursuit tasks. First, we compare the pursuit initiation behavior of the model with experimental data from studies on non-human primates (de Brouwer et al., 2002; Rasche and Gegenfurtner, 2009). Second, a predictive pursuit task of a sinusoidal target, where we evaluate the capability of the model to eliminate lag between target and eye velocities caused by sensory delays and perform predictive pursuit of occluded objects (Eckmiller and Mackeben, 1978; Whittaker and Eaholtz, 1982). Third, we evaluate the ability of the model to adapt to unpredictable perturbations and phase shifts of target velocity in experimentally observed timescales (Van den Berg, 1988).

In all the experiments, the RNN contains $N = 500$ neurons and they are fully connected. The initial recurrent weights are drawn from a Gaussian distribution with mean 0 and standard deviation $g/\sqrt{N}$ with $g = 1.5$, which results in a spontaneous chaotic behavior (Sussillo and Abbott, 2009). The readout weights are initialized to zeros. The feedback weights connecting the RNN neurons to the readout unit are drawn uniformly from the range -1 to

1 with gain $K = 1$. The integration timestep is 16 ms. The time constant τ is set to 160 ms and like previous models, the sensory delay δ is set to experimentally observed value of 80 ms (Krauzlis and Lisberger, 1994; de Xivry et al., 2013). For the leaky integrator, the time constant τ_l is equal to 128 ms. The values of α and K_l are set to 1.25 and 0.5, respectively, for the initiation experiment, and to 100 and 1, respectively, for the predictive pursuit experiment. The Matlab code used in the experiments is available at `https://github.com/hkashyap/predictivePursuit`.

Similar to (de Xivry et al., 2013), pursuit onset is detected by fitting a piecewise linear function (0 before pursuit onset T and $A(t-T)$ after T) to $\tilde{v}_E$ traces during an interval of 320 ms starting from stimulus onset. Similarly, for the initiation experiment, mean eye acceleration (B) is calculated by fitting $\tilde{v}_E$ traces during the interval 80-180 ms after pursuit onset to $\tilde{v}_E(T+0.08)+B(t-(T+0.08))$. The interval is selected to compare with experimental eye acceleration data (de Brouwer et al., 2002; de Xivry et al., 2013).

5.3.1 Pursuit Initiation

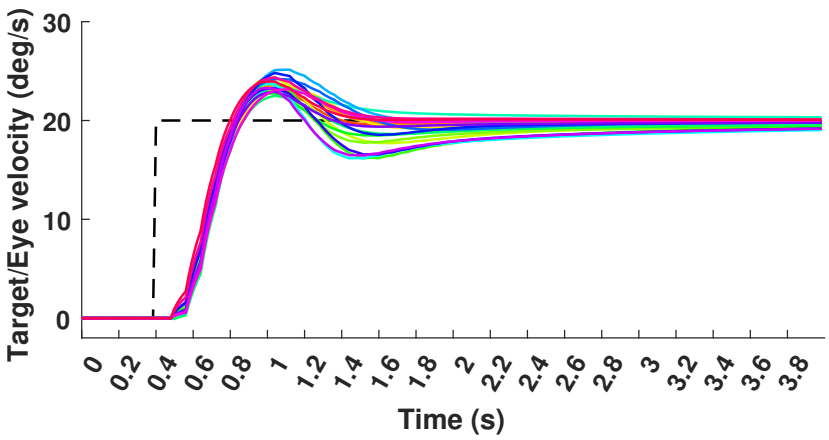


Figure 5.2: Eye velocity during pursuit initiation in response to a ramp stimulus of constant velocity 20 deg/s. The black dashed line depicts the target velocity. The colored lines are the eye velocity responses generated by the proposed model in 20 trials.

The smooth pursuit observed during sudden movement of a target with constant velocity af-

ter fixation, known as a ramp stimulus, has a very characteristic initial acceleration profile, as observed in human and non-human primate experiments (Robinson et al., 1986; Krauzlis and Lisberger, 1994; Spering and Gegenfurtner, 2007; Medina and Lisberger, 2009). Initiation is the most appropriate part of pursuit response for quantitative analysis, as it is consistent across subjects (Van den Berg, 1988). Figure 5.2 depicts the pursuit responses generated by our model for a ramp stimulus of velocity 20 deg/s. Similar to the experiments, the target is initially fixed. At 400 ms, the target suddenly starts moving with constant velocity 20 deg/s on a straight line. In all trials, our model generates the typical initial acceleration and the subsequent overshoot, comparable with experimental observations and outputs of the previous models (Robinson et al., 1986; Krauzlis and Lisberger, 1994; de Xivry et al., 2013). The pursuit response latency of our model from the onset of the stimulus is 146 ± 13.7 ms (mean±SD), which is similar to the average pursuit response latency of 150 ms measured experimentally for the ramp stimulus (Rasche and Gegenfurtner, 2009).

Consistent with experimental studies (Spering and Gegenfurtner, 2007; Medina and Lisberger, 2009; Rasche and Gegenfurtner, 2009), our model does not show an oscillatory behavior for the ramp stimulus after the overshoot. The recent predictive pursuit model by de Xivry et al. (2013) did not produce the oscillatory behavior either. Mainly early control theoretic pursuit models resulted in the oscillatory behavior (Krauzlis and Lisberger, 1994; Robinson et al., 1986). However, the pursuit model by Krauzlis and Lisberger (1994) employed a separate image-acceleration pathway, in addition to an image-velocity pathway, that generated the oscillatory behavior. Our model does not require separate mechanisms for pursuit acceleration and prediction.

We compare the mean eye acceleration during pursuit initiation generated by the proposed model versus experimental data of humans presented in (de Xivry et al., 2013), originally from a dataset by de Brouwer et al. (2002). Figure 5.3 depicts the comparison of mean eye acceleration profiles in response to targets velocities -50 deg/s to 50 deg/s at an increment of

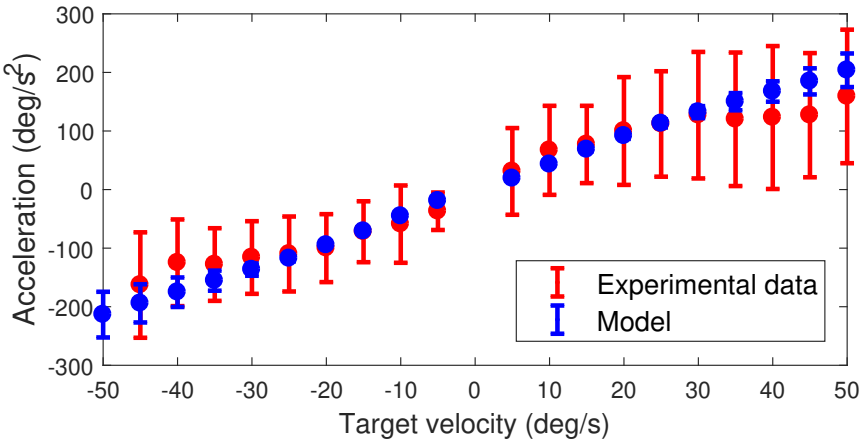
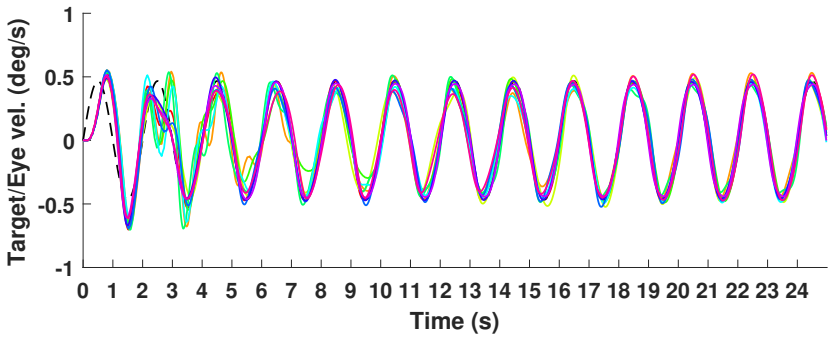


Figure 5.3: Mean eye acceleration versus target velocity between 80 ms and 180 ms after pursuit onset. Blue circles correspond to predictions by the proposed model and red circles correspond to experimental data by de Brouwer et al. (2002), reproduced from (de Xivry et al., 2013). Vertical bars are the standard deviations from mean. Experimental data is not available for target velocity -50 deg/s.

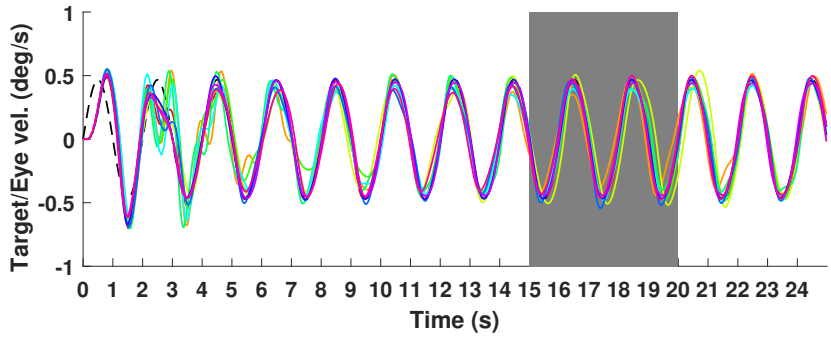
5 deg/s. Similar to (de Xivry et al., 2013), eye acceleration is calculated during the interval 80 ms to 180 ms after pursuit onset. The plot shows that acceleration generated by the proposed model during pursuit initiation follows a comparable trend as the experimental data. Similar to the experiment, the standard deviation of acceleration generated by our model gradually increases for higher target velocities, which is caused by large weight updates due to higher RS error signals. The mean acceleration produced by our model matches the experimental data for target velocities up to 30 deg/s. Beyond this range, the experimental data shows the effect of physiological limits as the acceleration plateaus. Similar to the previous models (Bennett and Barnes, 2006), a saturation function for acceleration may be used to reproduce this behavior.

5.3.2 Predictive Pursuit

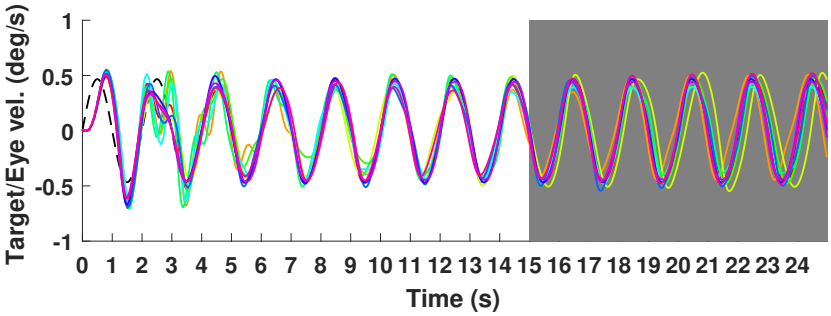
The visual system learns the target movement pattern when it is predictable (Barnes and Asselman, 1991; Deno et al., 1995; Whittaker and Eaholtz, 1982). The learned model allows



(a)



(b)



(c)

Figure 5.4: The pursuit eye velocity generated by our model in response to sinusoidal target velocity pattern. The black dashed line is the target velocity and the colored lines are the eye velocity simulated using the proposed model in different trials. The grey areas are the time periods where the target is occluded. (a) The target is always visible, (b) the target is temporarily occluded and then reappears, and (c) the target is permanently occluded after 15 seconds.

the eye to track the target with a smaller phase lag and during occlusion, which is not possible for unpredictable targets (Dallos and Jones, 1963; Yasui and Young, 1984; Barnes et al., 1987; Barnes and Ruddock, 1989). Similar to the existing predictive pursuit models (Shibata et al., 2005; de Xivry et al., 2013) and experiments (Van den Berg, 1988; Whittaker

and Eaholtz, 1982; Eckmiller and Mackeben, 1978; Keating, 1991), we test our model on a sinusoidally varying target velocity pattern. Primates track sinusoidal targets with little or no lag (Keating, 1991; Van den Berg, 1988).

Figure 5.4 shows the results of the experiment, which demonstrates the predictive capability of the proposed model in terms of almost zero lag pursuit and sustained tracking performance during occlusion. Figure 5.4a depicts the experiment where the target follows a sinusoidal velocity pattern with amplitude 0.47 deg/s and frequency 0.5 Hz. Since, the readout weights of the RNN are initialized to zero, the initial eye velocity is 0 deg/s. The retinal slip or error signal for learning is not available to the RNN during first 80 ms after target onset (sensory delay) and therefore, the eye lags behinds the target (evident from the target and the eye velocity plots in Figure 5.4a). Learning process starts after 80 ms and despite using a delayed error signal, it is able to eliminate the phase lag between the target and eye velocities within the first cycle of the sinusoid. Within a few cycles of the sinusoid, the eye velocity closely follows the target velocity. Primate experiments using periodic stimuli also observe that the phase error between target and eye velocities becomes small within the third cycle of sinusoid (Barnes, 2008; Van den Berg, 1988).

Figure 5.4b and Figure 5.4c illustrate the effect on pursuit performance due to temporary and permanent occlusion of the target, respectively, after the model has learned the target velocity pattern. In Figure 5.4b, when the target is occluded from 15 s to 20 s after target onset, the model continues to generate a eye velocity pattern that closely resembles the occluded target's velocity. Although, a phase error develops between the eye velocity and the target velocity during occlusion. After the target reappears, the phase error is corrected within a single cycle, much faster than the initial learning. Figure 5.4c shows that when the target is occluded permanently at 15 s after target onset, the model continues to generate sinusoidal eye velocity pattern for many cycles. However, the phase error between the target and eye velocity gradually increases. Similar experiments on humans and monkeys (Whittaker and

Eaholtz, 1982; Eckmiller and Mackeben, 1978) report that pursuit movement continues for a few cycles after a sinusoidal target is turned off and then a phase error develops gradually.

5.3.3 Unpredictable Perturbation and Phase Shift

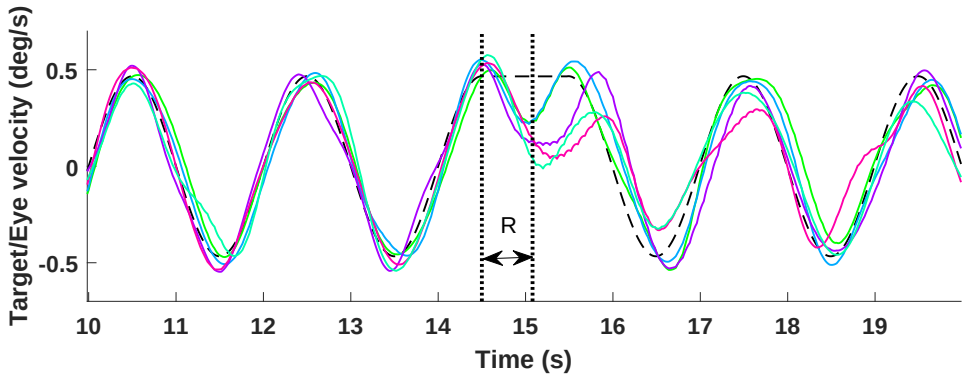


Figure 5.5: Response of the predictive pursuit model to unpredictable perturbation and phase shift. Black dashed line is the target velocity and the colored lines are the eye velocity generated by the model in 5 trials. R = 0.58 s is the experimental reaction time since perturbation calculated using the formula provided by Van den Berg (1988). Compares to Figure 8(a) of Van den Berg (1988). Pursuit starts at 0 s (not shown).

In human subject experiments, Van den Berg (1988) studied the effect of unpredictable perturbation and phase shift on predictive pursuit by replacing a sinusoidal velocity stimulus with a ramp stimulus for half cycle. During perturbations, the eye initially accelerated following the original course of the sinusoidal target before reversing acceleration to match the modified velocity. During this transition, the time at which the acceleration becomes zero since the beginning of perturbation is known as the reaction time (R). The study observed maximum reaction times when the target velocity was perturbed at the peak of the sinusoid, which were larger than one quarter of a cycle and approximated using the following formula.

$$R = \frac{1}{4 \times frequency} + 0.08s \tag{5.10}$$

Figure 5.5 depicts the eye velocity generated by our model during the same experiment, where perturbation occurs at the peak velocity of a 0.5 Hz sinusoidal target for half cycle. The reaction times achieved by our model in different trials are close to 0.58 s, the experimental reaction time obtained using Equation 5.10. Similar to the experimental data, our model adapts to the new phase within the first cycle after perturbation, and the phase error caused by the perturbation is corrected during the first two cycles after perturbation. The results from our model and experimental observations by Van den Berg (1988) show that the predictive pursuit system continuously learns the target velocity at a fast learning rate, which is not possible in memory based models.

5.3.4 Unpredictable Target Velocity

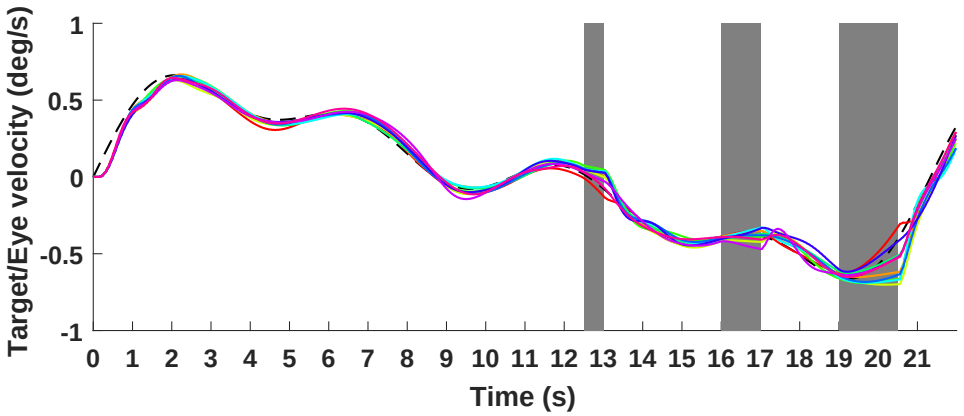


Figure 5.6: Eye velocity prediction by our model in response to an unpredictable target velocity input. The black dashed line is the target velocity sequence and the colored lines are the model output during ten trials. The grayed regions are occlusions.

We also tested the response of our model when the target velocity is not predictable. Similar to the human pursuit experiment for unpredictable targets (Collewijn and Tamminga, 1984), we use a pseudo-random target sequence that is a sum of four sine waves with different frequency and amplitude. The target velocity is not predictable, as evidenced by the deviation in pursuit prediction during the three blank periods, shown in Figure 5.6. Similar to the experimental observations by Collewijn and Tamminga (1984) for unpredictable targets, our

model is able to reduce the initial sensory lag using continuous prediction and then switches between phase lead and phase lag to maintain small prediction error. The third occlusion at 19 s clearly illustrates the unpredictability of the target movement, as the model expected the target velocity to either plateau, increase, or decrease. The deviations caused by target blanking are corrected after target reappearance.

5.4 Discussion

The RNN model presented in this chapter generates the predictive behaviors observed in human and non-human primate smooth pursuit experiments. It is able to achieve almost zero lag tracking of sinusoidal targets by eliminating sensory delays, track an occluded target with a non linear velocity profile, and adapt to unpredictable perturbation and phase shift of target velocity in experimentally observed timescales. The model also qualitatively reproduces the experimentally observed initial pursuit acceleration. To the best of our knowledge, this is the first neural network model to achieve all of the above mentioned smooth pursuit behaviors. It demonstrates that a single neural network can generate pursuit initiation dynamics and persistent predictive pursuit signals. Although, pursuit experiments on primates previously suggested that an internal model of target motion may be used for pursuit prediction (Barnes, 2008; Whittaker and Eaholtz, 1982; Eckmiller and Mackeben, 1978; Van den Berg, 1988), the neural mechanism to create and maintain the internal model was not known. Our work shows how the internal model is learned and updated rapidly by an RNN using a delayed RS signal as error, in order to reduce tracking lag, generate persistent pursuit during occlusions, and correct eye velocity during target perturbations.

A puzzling aspect of smooth pursuit eye movement is that during zero lag tracking and occlusion, pursuit movement continues when RS is zero. Figure 5.7 depicts the RS signal at the retina (without delay) during pursuit of a sinusoidal target by our model. It can

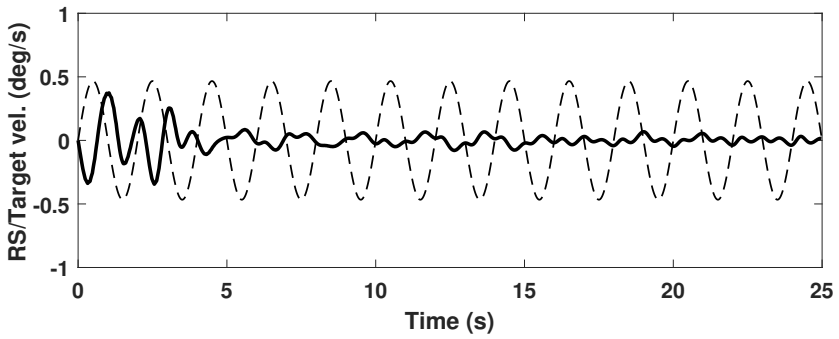


Figure 5.7: Mean RS (the solid black trace) from 10 trials of the experiment shown in Figure 5.4a. The target velocity (the dashed line) is superimposed for reference. The RS signal is received by the predictive model after 80 ms to simulate sensory delays.

be seen that after the target motion pattern is learned, the RS is not exactly zero, but deviates by small amounts around zero. However, the small RS components cannot drive pursuit output during occlusion. We propose that the RNN is able to generate self-sustained eye velocity predictions. The small RS components are continuously used to correct the pursuit prediction. During occlusion, these corrective RS components are not available, and therefore, the pursuit eye velocity gradually lags behind the target, similar to experimental data (Whittaker and Eaholtz, 1982). In our model, the RNN operates in a chaotic regime and each neuron has its own baseline spontaneous activity. During occlusion, the learned neural activity pattern continues to produce the pursuit prediction.

5.4.1 Other Computational Models of Smooth Pursuit

Early pursuit models implemented a feedforward controller that canceled out efferent feedback to achieve high velocity gain (Robinson et al., 1986; Yasui and Young, 1984). Krauzlis and Lisberger (1994) proposed a feedback control model using parallel velocity and acceleration pathways with second order filters to process RS. This model used under-damped filters to generate the ringing behavior of the earlier models (Robinson et al., 1986). None of these models achieved zero lag pursuit of a periodic signal because they relied on a delayed

RS information.

From control-theoretic perspective, the current target velocity information is required to predict current eye velocity without lag. To work around this, later models used prior knowledge or memory of target motion to estimate the current target velocity. Bahill and McDonald (1983) proposed a model for generating pursuit eye movements based on *a priori* knowledge of target trajectory. Other memory based models proposed to generate pursuit eye movements based on stored patterns for periodic trajectories (Barnes and Wells, 1999). de Xivry et al. (2013) proposed a memory based model that used the target trajectory stored from prior trials to run a Kalman filter for eye velocity prediction. However, as indicated by Shibata et al. (2005), memory based models are biologically not plausible, because i) they require a periodicity estimator in the brain, ii) the improvement in pursuit lag by memory-based models can only be periodic, whereas studies (Van den Berg, 1988; Keating, 1991) found a rather gradual decrease in lag between target and eye velocities, and iii) they cannot adapt to unpredictable perturbations of a periodic signal within a single cycle as observed in humans.

Shibata et al. (2005) removed the requirement for prior knowledge of target trajectory and adapted the parameters of a Kalman filter online for eye movement predictions. However, these Kalman filter based approaches have drawbacks, i) they cannot account for acceleration/deacceleration caused by the external target and operate using the negative feedback of prediction error and ii) filter state and prediction are static during occlusions. Both drawbacks arise since these models do not learn the target motion pattern. Moreover, the model is not tested for long occlusions (maximum tested occlusion is 1/10th of a cycle).

5.4.2 The role of FEF in Predictive Pursuit

Many studies observed a direct role of FEF in predictive smooth pursuit generation. Keating (1991) found that lesions or ablations of FEF impaired monkeys' ability to conduct smooth pursuit of sinusoidal targets when the target was visible and during occlusions. The study reported that the pursuit lag between eye and target increased from ≈7 ms before ablation to ≈100 ms after ablation. Single unit recordings in FEF found neurons that continued to respond strongly after a sinusoidal target had been extinguished (MacAvoy et al., 1991). Fukushima et al. (2002) found similar FEF neurons from recording studies on monkeys. Predictive pursuit eye movement signals in FEF were also observed during fMRI studies (Lencer et al., 2004). These studies suggest that FEF learns an internal model of the target velocity pattern to signal predictive pursuit eye movements, regardless of whether the target is visible or not. This implies that FEF is a plausible neural correlate for our RNN, based on their common predictive activities during smooth pursuit and location on the pursuit pathway. Whereas, the leaky integrator can be realized in dorsal pontine nuclei (PN) and reticularis tegmenti pontis (NRTP) regions in the brainstem as they relay FEF output to cerebellum.

Chapter 6

A Fully Neuromorphic Stereo Vision System for Dynamic scenes

6.1 Introduction

Marr and Poggio (1976) proposed that complex information processing systems, such as the visual cortex in the brain, should be understood at three distinct and complementary levels of analysis, which are computational, algorithmic, and implementation levels. These three levels are often considered in cognitive sciences for studying brain computations (Dawson, 1998). At the computational level, systems are specified as to what problems they solve. At the algorithmic level, systems are understood by considering how information is represented and the processes used to manipulate these representations to solve a computational problem. At the implementation level, systems are described in terms of the physical substrate used to realize computation, such as neural structures in the brain or in silicon using transistors. The computational vision models presented in Chapter 3, Chapter 4 and Chapter 5, are all examples of algorithmic level analysis, since they provide the computations required to solve

a problem, however do not specify how to implement those in hardware or how the cortical circuits realize the representations.

In this chapter, we present a stereo vision system at the implementation level, in terms of the silicon circuit elements that process external visual input at multiple stages of representation. The elements of this vision system are inspired by the biological neural circuits and in particular, follow two important information processing principles of the brain, viz., sparsity of representation and parallel asynchronous computation. These principles are implemented to solve complex computations while using a tiny fraction of the energy consumed by stored-program computers (Neumann, 1958).

While the artificial neural networks may not operate the same way as the brain, they utilize highly parallel and hierarchical architectures that gradually abstract input data to more meaningful concepts (Bengio et al., 2013; Riesenhuber and Poggio, 1999; DiCarlo et al., 2012). On the other hand, sparse and asynchronous computation has not been equally adopted yet, since the bulk of machine learning applications currently run on remote clusters or supercomputers with a huge energy budget. Nevertheless, for autonomous vehicles, drones, robots, satellites, and the imminent smart edge devices, energy consumption is a challenge (Beard et al., 2005). Another barrier for sparse and asynchronous computation are the traditional sensors, such as frame-based cameras, which provide dense inputs to algorithms in use.

However, recently developed event-based cameras (Lichtsteiner et al., 2006; Brandli et al., 2014; Posch et al., 2011), inspired by the biological retina, encode pixel illumination changes as asynchronous events at high temporal resolution. These sensors, also known as silicon retinas, address two major drawbacks of using frame-based cameras for real-time applications. First, throughput of frame-based applications is limited by the camera frame rate, usually 30 or 60 frames per second. Event-based cameras generate events at microsecond resolution. Second, consecutive frames in videos are usually highly redundant, which waste downstream data transfer and computing resources. On the other hand, events are sparse

and non-redundant, leading to optimal downstream resource usage as per actual need. Moreover, event-based cameras have high dynamic range in the order of 100 dB, which is ideal for real world variations in lighting conditions.

To achieve low energy and real-time benefits of event-based inputs, computations must be performed asynchronously as they arrive, similar to the neural operations in the brain. Traditional computing platforms, such as CPUs and GPUs, are clock-driven, i.e. computation datapaths are executed in each cycle even without any new data. As a result, these processors cannot benefit from asynchronous and sparse representation of event data.

To benefit from sparse and asynchronous computation using event-based sensor data, neuromorphic processors have been developed recently (Merolla et al., 2014; Furber et al., 2013; Indiveri et al., 2006; Benjamin et al., 2014; Schemmel et al., 2010). These processors represent input events as spikes of neuron membrane potential and process them in parallel using a large population of neurons. They are stimulus-driven, i.e. a neuron computes when it receives an input spike. Also, the propagation delay of an event through the neuron layers is usually a few milliseconds. This and other brain-inspired design principles, such as localized memory and computation, allow for low power and real-time cognitive applications.

Previously, neuromorphic processors were used successfully for real-time CNNs (Esser et al., 2016), interactive character recognition (Sawada et al., 2016), optic flow (Brosch and Neumann, 2016), and gesture recognition (Amir et al., 2017). Another important vision task for freely navigating autonomous mobile agents is depth perception. Lidar depth sensors used in autonomous vehicle experiments are highly expensive (Belbachir et al., 2014) and regular frame-based cameras are not robust to real world variations in motion and lighting conditions. The speed and low power requirements of these applications can be effectively met using event-based sensors. Event-based stereo vision provides additional advantages over other depth estimation methods that increase accuracy and save energy, such as high temporal resolution, high dynamic range, inbuilt foreground extraction, and robustness to

interference with other agents.

Several methods have been proposed to solve event-based stereo correspondence. Most global methods (Mahowald, 1992; Dikov et al., 2017; Piatkowska et al., 2017; Osswald et al., 2017) are derived from the cooperative stereo algorithm by Marr et al. (1976). The algorithm assumes depth continuity and the event-based implementations are not tested with objects tilted in depth. Local methods can be parallelized and they find corresponding events using either local features over a spatiotemporal window or event-to-event features (Camuñas-Mesa et al., 2014; Schraml et al., 2010; Rogister et al., 2012; Kogler et al., 2011; Schraml et al., 2016). However, most approaches are proof-of-concept and implemented using non-event-based hardware, such as in a CPU or a DSP.

In this chapter, we propose a local event-based stereo disparity algorithm using multiscale spatiotemporal features and its fully neuromorphic implementation that allows to calculate disparities at 2,000 frames per second, ideal for use with live high speed event cameras, such as DVS (Lichtsteiner et al., 2006). The main contributions of the proposed method, regarding the related state of the art (Dikov et al., 2017; Piatkowska et al., 2017; Osswald et al., 2017; Rogister et al., 2012; Schraml et al., 2016), are multiscale matching, end-to-end neuromorphic disparity calculation, high throughput and low latency (9-22 ms), robustness to depth gradients and high speed moving objects, and linear scalability to multiple neuromorphic processors for larger input sizes. The proposed method is ideal for autonomous vehicles and robots to calculate depth of moving objects in outdoor scenes in real-time using only a few hundred mW.

This chapter is based on previously published work:

Andreopoulos, A.*, Kashyap, H. J.*, Nayak, T. K., Amir, A., & Flickner, M. D. (2018). "A low power, high throughput, fully event-based stereo system". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 7532-7542). (* equal

contribution)

Portions are reprinted with permission, © 2018 IEEE.

6.2 Neuromorphic Hardware

6.2.1 Dynamic Vision Sensors

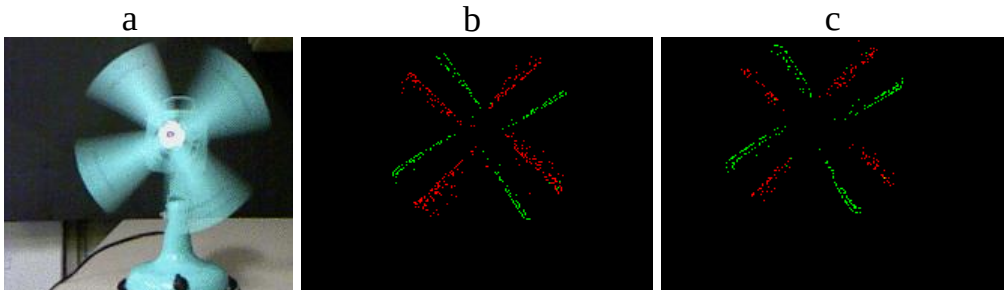


Figure 6.1: Frame based (a) and event-based (b-left DAVIS and c-right DAVIS) camera output for a rotating fan. Green dots are ON events, i.e. an increase in pixel intensity, and red dots are OFF events.

The biological retina works in a more effective way than a regular frame-based camera. Rather than capturing the pixel intensity values throughout the whole visual scene at all times, it only captures the changes in the intensity values. This is more efficient in terms of resources required for processing visual data than regular frame-based processing. Mahowald (1992) proposed an analog VLSI circuit to capture this characteristic and termed it as "Silicon retina". Her work paved the way for more variants of event-based sensors, such as DVS (Lichtsteiner et al., 2008) and DAVIS cameras (Brandli et al., 2014).

Event-based dynamic vision sensors record the intensity changes above a certain threshold at each pixel. When the intensity of a pixel increases above a certain threshold, an ON event is registered. Similarly, when the intensity of the pixel decreases more than a certain other threshold, an OFF event is registered. Therefore, these sensors do not produce any output for

a static scene. This improves speed, power, dynamic range, and computational requirements. The events or spikes are produced in an address-event representation (AER) format, which contains the location of the pixel producing the spike and the time-stamp. Event-based sensors provide high temporal resolution and consume very low energy. For example, DAVIS provides microsecond level timing precision and latency, a large output bandwidth of 50 million events/sec, and consumes maximum 14 mW power (Brandli et al., 2014). The high temporal resolution of DAVIS events provides more features of high speed objects for stereo matching than a frame-based camera. This increases accuracy in comparison to frame-based input, where high speed blurs object features, as shown in Figure 6.1.

6.2.2 TrueNorth Processor

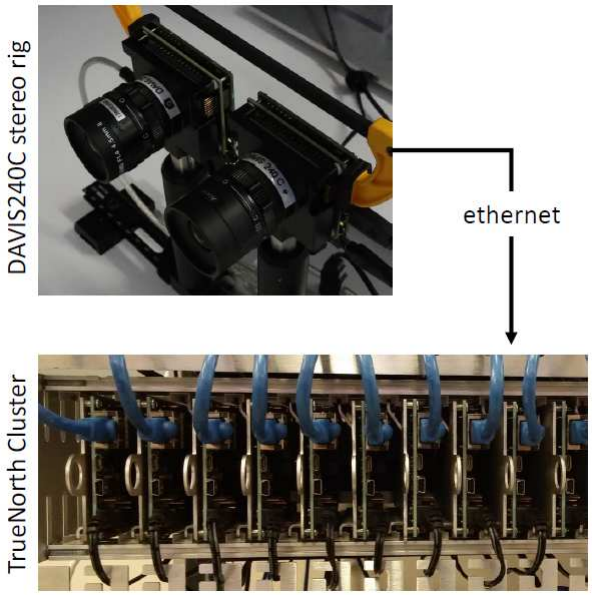


Figure 6.2: The time-stamp synchronized stereo rig is connected to a cluster of TrueNorth chips via ethernet.

Our implementation uses a pair of synchronized DAVIS240C cameras, connected via Ethernet to a cluster of TrueNorth NS1e boards (Figure 6.2). The IBM TrueNorth chip is a reconfigurable, non-von Neumann processor containing 1 million spiking neurons and 256 million

synapses distributed across 4096 parallel, event-driven, neurosynaptic cores (Merolla et al., 2014). Cores are tiled in a 64×64 array, embedded in a fully asynchronous network-on-chip. Under normal workloads, the chip consumes 70mW when operating at a 1 ms computation tick. Depending on event dynamics and network architecture, chip overclocking is possible, in which we can achieve as low as 0.5 ms per tick, thus doubling the maximum frame-rate achievable to 2000 ticks per second. The present work uses this overclocking method. Each neurosynaptic core connects 256 inputs to 256 neurons using a crossbar of 256×256 synapses with 8 bits of weight precision, plus a sign bit. A neuron state variable called a membrane potential integrates synaptically weighted input events with an optional leak decay. Neurons can be configured to either generate an output event deterministically, whenever the membrane potential $V(t)$ exceeds a threshold; or stochastically, with a pseudorandom probability related to the difference between the membrane potential and its threshold (Cassidy et al., 2013). The membrane potential is updated at each tick t to $V(t) = V(t-1) + \frac{\partial V(t)}{\partial t}$, followed by the application of an activation function $\mathbf{a}_n(V(t))$ where

$$\mathbf{a}_n(V(t)) = \begin{cases} 1, & \text{if } V(t) \geq n \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

Each neuron is assigned an initial membrane potential V(0). Furthermore, upon producing an event, a neuron is reset to a user-specified value. Unless specified otherwise, we assume initial membrane potentials and reset values of zero. TrueNorth programs are written in the Corelet Programming Language, a hierarchical, compositional, object-oriented language (Amir et al., 2013).

6.3 Methods

The proposed event-based stereo correspondence algorithm is implemented end-to-end as a neuromorphic system. This consists of systems of equations defining the behavior of TrueNorth neurons, encased in modules called corelets (Amir et al., 2013), and the subsequent composition of the inputs and outputs of these modules. Figure 6.3 depicts the sequence of operations performed by the corelets using inputs from stereo event sensors.

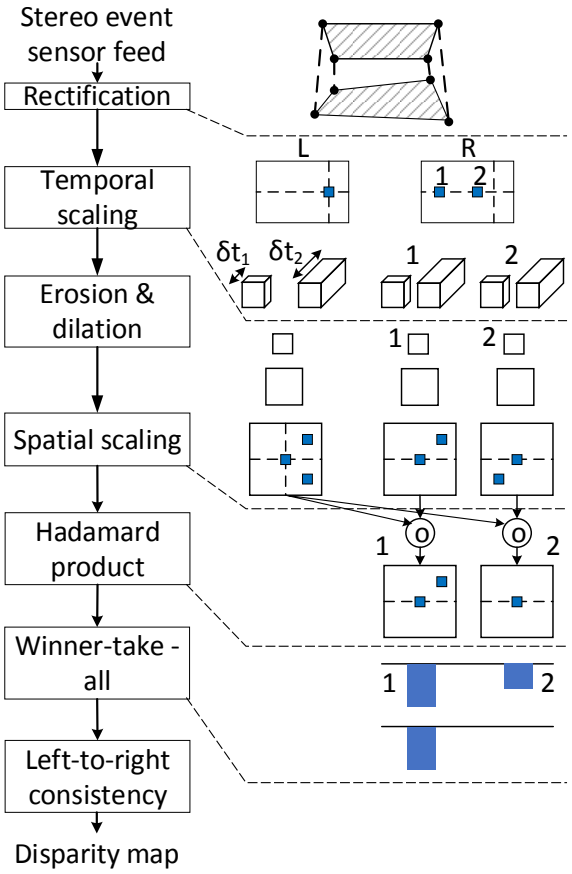


Figure 6.3: The pipeline of execution using input events generated by left and right sensors. A toy example of main operations performed is demonstrated side-by-side in a single spatiotemporal scale, with a event on the left image and its two candidate corresponding events on the right image. Standard morphological operations and left-to-right consistency check are not demonstrated.

6.3.1 Rectification

The stereo rectification is defined by a pair of functions $\mathcal{L}$, $\mathcal{R}$ which maps each pixel in the left and right sensor's rectified space to a pixel in the left and right sensor's native resolution respectively. On TrueNorth, this is implemented using $|H| \cdot |W|$ splitter neurons per sensor & polarity channel, arranged in an $|H| \times |W|$ retinotopic map. The events at each rectified pixel $p \in H \times W \times \{\mathcal{L}, \mathcal{R}\} \times \{+, -, \{+, -\}\}$ are generated through splitter neurons which replicate corresponding sensor pixels. Their membrane potential $V_p^{spl}(t)$ is defined by $\frac{\partial V_p^{spl}(t)}{\partial t} = I(t-1; p')$ where $I(t; p') \rightarrow \{0, 1\}$ denotes whether a sensor event is produced at time t and the sensor pixel p' corresponding to the mapping function in p. $\mathbf{a}_1(V_p^{spl}(t))$ defines the activation of the corresponding neuron. Potentials are initialized to zero and set to also reset to zero upon spiking. On TrueNorth, temporal variable t quantizations of up to 2,000Hz are achievable.

6.3.2 Multiscale Temporal Representation

The event rate of an event-based sensor depends on factors, such as scene contrast, sensor bias parameters, and object velocity. To add invariance across event rates, we accumulate spikes over various temporal scales through the use of temporally overlapping sliding windows. These temporal scales are implemented through the use of splitter neurons and a temporal ring buffer mechanism, which cause each event to appear at its corresponding pixel multiple times. The ring buffer is implemented by storing events in membrane potentials of memory cell neurons in a circular buffer, and through the use of control neurons which spike periodically to polarize appropriate memory cell neurons.

A control neuron that produces events with period T and phase ϕ is defined by $\mathbf{a}_T(V_\phi^{ctrl})$ that satisfies $\frac{\partial V_\phi^{ctrl}(t)}{\partial t} = 1$, $V(0) = \phi$ and resets to zero upon producing an event. Through populations of such neurons one can also define $\mathbf{a}_T(V_{[\phi,\theta]}^{ctrl})$ corresponding to phase intervals

$[\phi, \theta]$ (where $\theta - \phi + 1 \leq T$), during which events are periodically produced. Control neurons are used to probe (prb) or reset (rst) membrane potentials of memory cell neurons.

A memory cell neuron receives its own output via a recurrent connection, input axons to set the membrane potential value, and control axons for resetting and querying the memory cell. The output at index $r \in \{1, ..., T+2\}$ of a $T+2$ size memory cell ring-buffer at a given pixel p, is multiplexed via two copies ($m \in \{0, 1\}$) and is denoted as $\mathbf{a}_2(V^{mem}_{p,m,r})$, where $\hat{r} = t \bmod (T+2)$. The membrane potential dynamics of memory cell neurons are defined as,

$$\begin{aligned}
&\frac{\partial V^{mem}_{p,m,r}(t+1)}{\partial t} = [[\mathbf{a}_1(V^{spl}_p(t))]^r_{\hat{r}} + \mathbf{a}_2(V^{mem}_{p,m,r}(t-1)) \\
&-[\mathbf{a}_T(V^{rst}_{[r,T+r-1]}(t))]^m_1 - [\mathbf{a}_T(V^{rst}_{[r,T+r-1]}(t))]^{m+1}_1 \\
&+[\mathbf{a}_T(V^{prb}_{[1+r,T+r]}(t))]^m_1 + [\mathbf{a}_T(V^{prb}_{[1+r,T+r]}(t))]^{m+1}_1 \\
&-\mathbf{a}_{T+2}(V^{rst}_{r-1}(t))]_+
\end{aligned} \tag{6.2}$$

where probe/reset (prb/rst) control neurons are used,

$$[\mathbf{x}]^r_{\hat{r}} = \begin{cases} \max\{0, \mathbf{x}\}, & \text{if } r = \hat{r} \\ 0, & \text{otherwise} \end{cases} \tag{6.3}$$

and $[\mathbf{x}]_+ \stackrel{\text{def}}{=} [\mathbf{x}]^1_1$ denotes a ReLU function. Eq. 6.2 defines a ring-buffer with $T+2$ memory cells, where probe pulses periodically and uniformly query T of the $T+2$ cells for the stored memory contents at each tick, where $m = 0$ neurons are probed at even ticks and $m = 1$ neurons are probed at odd ticks. Memory cell neurons, in conjunction with appropriately timed control neurons define a ring buffer mechanism to encode input events at various timescales.

6.3.3 Morphological Erosion and Dilation

Binary morphological erosion and dilation is applied to denoise the image. Given a 2-D neighborhood $N(p)$ centered around each pixel p, the erosion neuron's membrane potential V_p^e is guided by following equations,

$$\frac{\partial V_p^e(t)}{\partial t} = [1 - |N(p)| + \sum_{q \in N(p)} \sum_m \sum_r \mathbf{a}_2(V_{q,m,r}^{mem}(t-1))]_+ \tag{6.4}$$

and uses an $\mathbf{a}_1$ activation function. Similarly, dilation neurons V_p^d with receptive fields $N(p)$ evolve according to

$$\frac{\partial V_p^d(t)}{\partial t} = \sum_{q \in N(p)} \mathbf{a}_1(V_q^e(t-1)) \tag{6.5}$$

The neuron potentials are initialized to zero and also reset to zero upon producing a spike. Cascades of erosion neurons followed by dilation neurons, are used to denoise retinotopic event inputs, as well as to regularize disparity maps, applied to each disparity level's binary map.

6.3.4 Multiscale Spatiotemporal Features

Each feature extracted around a pixel p in the left or right sensor, is a concatenation of event patches, extracted at different spatiotemporal scales. In contrast to temporal scaling described in Section 6.3.2, spatial scaling consists of sampling at different resolutions. This results in a set of spatiotemporal coordinate tensors $\mathcal{X}_{L,p}$, $\mathcal{X}_{R,p}$ which define the coordinates where events form feature vectors at each time step t. The i^{th} of these coordinates is represented by neuron activations $\mathbf{a}_1(V^{L^{\{+,-\}}}_{\mathcal{X}^{(i)}_{L,p}}(t))$ and $\mathbf{a}_1(V^{R^{\{+,-\}}}_{\mathcal{X}^{(i)}_{R,p}}(t))$ in the left and right

sensor's positive (+) or negative (-) polarity channel. For brevity, we henceforth drop the $+, -$ superscripts for distinct event streams based on polarity.

6.3.5 Hadamard Product

Given a pair of spatiotemporal coordinate tensors $\mathcal{X}_{L,p}$, $\mathcal{X}_{R,q}$ centered at coordinates p, q in the left and right sensor respectively and representing K coordinates each, we calculate the binary Hadamard product $\mathbf{f}_L(p,t) \cdot \mathbf{f}_R(q,t)$ associated with the corresponding patches at time t, where $\mathbf{f}_L(p,t) = \prod_i \{\mathbf{a}_1(V^L_{\mathcal{X}^{(i)}_{L,p}}(t))\} \in \{0,1\}^K$ and $\mathbf{f}_R(q,t) = \prod_i \{\mathbf{a}_1(V^R_{\mathcal{X}^{(i)}_{R,q}}(t))\} \in \{0,1\}^K$. The product is calculated in parallel across multiple neurons, as K pair-wise logical AND operations of corresponding feature vector entries, resulting in $(\mathbf{a}_1(V^{dot}_{p,q,1}), ..., \mathbf{a}_1(V^{dot}_{p,q,K}))$ where

$$\frac{\partial V^{dot}_{p,q,i}(t)}{\partial t} = [\mathbf{a}_1(V^L_{\mathcal{X}^{(i)}_{L,p}}(t-1)) + \mathbf{a}_1(V^R_{\mathcal{X}^{(i)}_{R,q}}(t-1)) - 1]_+ \tag{6.6}$$

This population code representation of the Hadamard product output is converted to a thermometer code (Kak, 2016), which is passed to the winner-take-all circuit described below, which determines the highest scoring disparity value. [1]

6.3.6 Winner-take-all

The winner-take-all (WTA) algorithm takes as input the thermometer codes of Hadamard product results in the previous step for D distinct candidate disparity levels and finds the disparity with the largest Hadamard product. The WTA algorithm consists of two steps: (i) the conversion of a thermometer code of input events representing an integer value, to a more efficient neuron-wise base-4 representation and (ii) the subsequent processing to

[1] e.g., given a population code $(0,1,0,1,1)$ representing value 3, its thermometer code is the juxtaposition of all events: $(1,1,1,0,0)$.

determine which of the base-4 values is the largest. This is a critical component of the system to ensure maximal throughput (frames-per-second) and hence to also minimize the active power consumption.

We assume a maximum thermometer length of $4^{B+1} \geq K$ for some $B \in \mathbb{N}$. Then for any $a \in \{0, 1, 2\}$, $b \in \{0, 1, ..., B\}$ we define the conversion of the $d \in \{1, ..., D\}$ candidate disparity to a base-4 membrane potential $V^{CNV}_{a,b,d}(t)$ as

$$\frac{\partial V^{CNV}_{a,b,d}(t)}{\partial t} = [\, -3 \sum_{i \in \overline{U(b)}} v^{t-1}_d(4^b \cdot i) + \sum_{i \in U(b)} v^{t-1}_d(4^b \cdot i) - a]_+ \tag{6.7}$$

where function v^t_d is the thermometer code representation of the d^{th} disparity/dot-product calculated at tick t, $\overline{U(b)} = \{x : 1 \leq x \leq 4^{B-b} - 1\}$ and $U(b) = \{x : 1 \leq x \leq 4^{B-b+1} - 1\} \setminus \overline{U(b)}$. All the conversion neurons use an $\mathbf{a}_1$ activation function and reset to 0 membrane potential upon spiking. Notice that $(\mathbf{a}_1(V^{CNV}_{0,b,d}(t)), \mathbf{a}_1(V^{CNV}_{1,b,d}(t)), \mathbf{a}_1(V^{CNV}_{2,b,d}(t)))$ is a length-3 thermometer code representation of a value in $\{0, 1, 2, 3\}$, representing the b^{th} digit in the base-4 representation of the value represented by v^{t-1}_d.

Given the base-4 representation of the inputs, the WTA consists of a pruning process where we iteratively process the $B+1$ components of each input, starting from the most signification base-4 values, to prune the thermometer codes not equal to the maximum. The membrane potential $V^{WTA}_{b,d}$ of stage b and disparity index d is given by,

$$\frac{\partial V^{WTA}_{b,d}(t)}{\partial t} = [R + \sum_{a=0}^{2} [\mathbf{a}_1(V^{CNV}_{a,b,d}(t-1)) - \max_k \{\mathbf{a}_1(V^{CNV}_{a,b,d}(t-1))\}]]_+ \tag{6.8}$$

where, R is

$$R = \begin{cases} \mathbf{a}_0(V_{b-1,d}^{WTA}(t-1)) - 1, & \text{if } b > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.9}$$

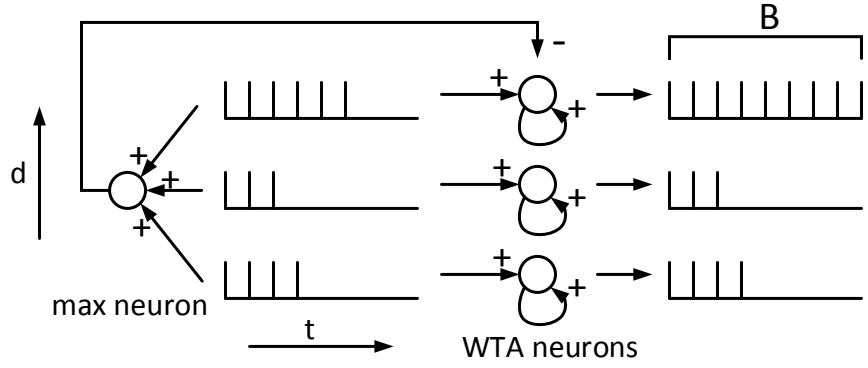


Figure 6.4: The WTA circuit and an example of operation. The bias -1 and control signals in Equation 6.9 are not shown.

Figure 6.4 depicts an example of operation of the WTA circuit. Each WTA neuron uses $\mathbf{a}_0$ activation function. Therefore, during step $b = 0$, WTA neurons are not pruned (i.e. producing an event) if input is equal to the maximum value. For subsequent steps $b > 0$, the output is not pruned iff, (i) it was not pruned in the previous step (the $\mathbf{a}_0(V_{b-1,d}^{WTA}(t-1)) - 1$ component of Equation 6.9) and (ii) the input at step b is equal to the maximum value. The elegant aspect of this WTA algorithm is that it enables single-tick WTA to take place. The conversion of the data to a base-4 representation accounts for the maximum fan-in of 256 per neuron on TrueNorth and decreases the $B + 1$ layers over which we pipeline, leading to lower system latency.

6.3.7 Consistency Constraints

A left-right consistency check is then performed to verify that for each left-rectified pixel p matched to right-rectified pixel q, it is also the case that right-rectified pixel q gets matched to left-rectified pixel p. This is achieved using two parallel WTA streams. Stream 1 calculates the winner disparities for left-to-right matching, and stream 2 calculates the winner disparities of right-to-left matching. The outputs of each stream are represented by D retinotopic maps expressed in a fixed resolution ($\mathbf{D}^{v}_{i,j,d}(t)$, $d \in \{0, ..., D-1\}$, $v \in \{L, R\}$), where events represent the retinotopic winner disparities for that stream. The streams are then merged to produce the disparity map $\mathbf{D}^{L,R}_{i,j,d}(t) = \mathbf{a}_1(V^{L,R}_{i,j,d}(t))$ where

$$\frac{\partial V^{L,R}_{i,j,d}(t)}{\partial t} = [\mathbf{D}^{L}_{i,j,d}(t-1) + \mathbf{D}^{R}_{i,j-d,d}(t-1) - \mathbf{a}_1(V^{spl}_{(i,j,\mathcal{L},\cdot)}(t-\hat{t})) - 2]_{+} \tag{6.10}$$

where $\hat{t}$ is the propagation delay of the first layer splitter output events until the left-right consistency constraint merging takes place. This enforces that an output disparity is produced at time-stamp t and pixel (i, j) only for left-rectified pixel (i, j), where an event was produced at $t - \hat{t}$.

6.4 Experimental Results

6.4.1 Datasets

We evaluate the performance of the system on sequences of random dot stereograms (RDS) representing a rotating synthetic 3D object (Figure 6.5a-f), and two real world sequences, consisting of a fast rotating fan (Figure 6.5g-m) and a rotating toy butterfly (Figure 6.5n-y) captured using the DAVIS stereo cameras. The synthetic dataset provides dense disparity estimates, which are not feasible to acquire with the sparse event based cameras. The dataset

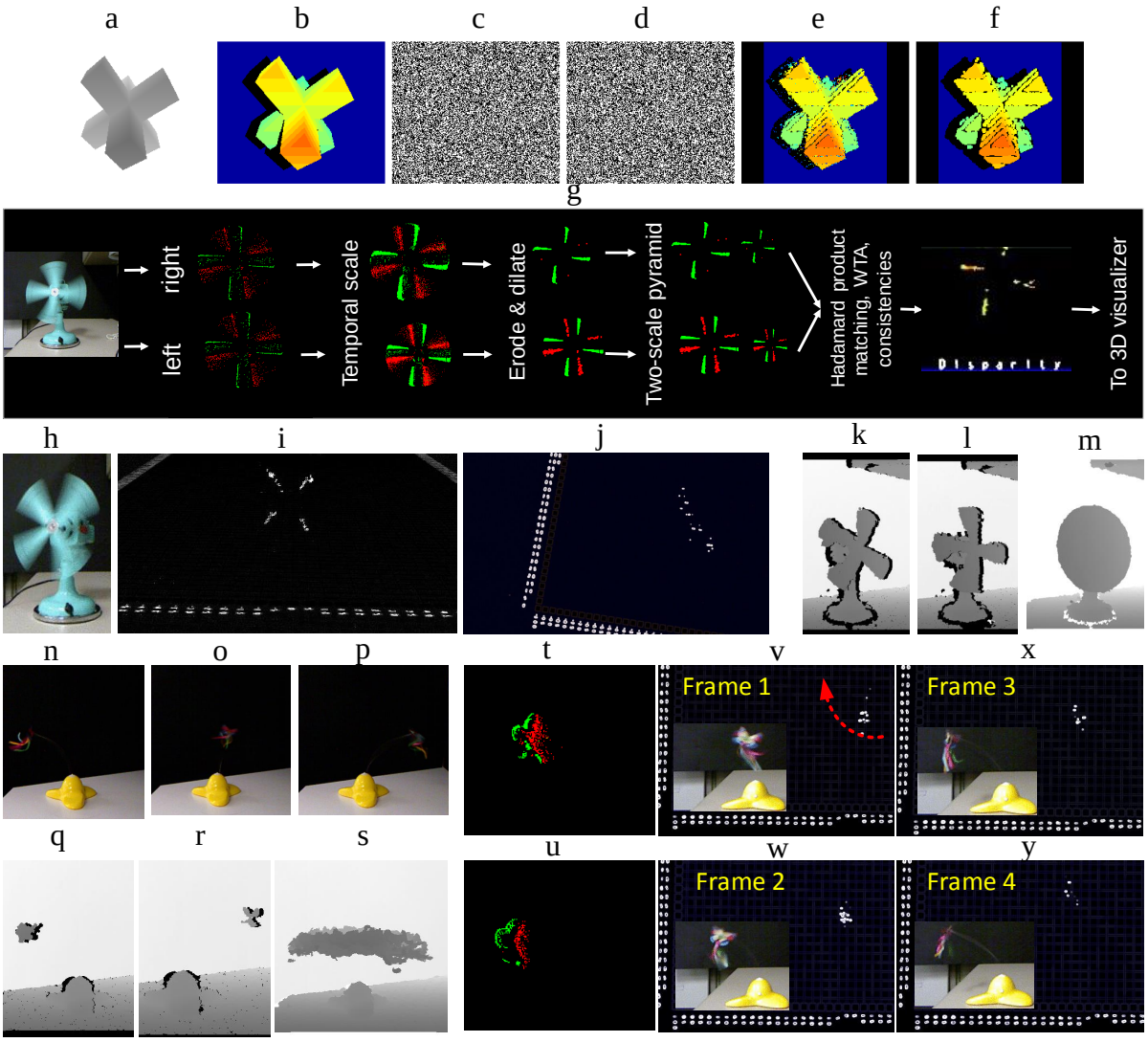


Figure 6.5: Experimental results obtained using TrueNorth. a) Example synthetic depth pattern, b) ground truth depth map, c) random dot stimuli (RDS), d) depth map superimposed on RDS, e) depth map obtained from corelet implementation, f) corelet result after erosion and dilation, g) fan sequence input received from the left-right DAVIS cameras and results generated by each layer of corelets from this input, h) example frame with fan rotating in a particular orientation, i) 3D reconstruction by the proposed method as seen from an angled front view (screen capture from the 3D visualizer), j) 3D reconstruction from a top view, k-l) Kinect depth maps with static fan blades, m) merged Kinect depth map, n-p) butterfly rotating around the spring base, q-r) Kinect depth map for the butterfly frames, s) merged Kinect depth maps, t) left DAVIS output for a 3 ms time window, u) right DAVIS output during the same window, v-y) top view from the 3D visualizer of 3D reconstruction four consecutive frames in the sequence at the butterfly rotates clockwise

is generated by assigning to each left sensor pixel a random event with a 50% probability. Similarly, each right sensor pixel is assigned a value by projecting it to the 3D scene and re-projecting the corresponding data-point to the left camera coordinate frame to find the closest pixel value. Self-occluded pixels are assigned random values. We measure the average disparity error, and the average recall, which is defined as the fraction of pixels where a disparity measurement was found.

For the non-synthetic datasets, a Kinect (Zhang, 2012) is used to extract ground truth of the scene structure. This also entails a calibration process for transforming the undistorted Kinect coordinate frame to the undistorted DAVIS sensor coordinate frame. Performance is measured in terms of precision, which is defined as the median relative error $\frac{\|x-x'\|}{\|x'\|}$ between each 3D coordinate x extracted in the DAVIS frame using the neuromorphic algorithm, and the corresponding ground coordinate x' in the aligned Kinect coordinate frame. Performance is also reported in terms of the recall, defined herein as the percentage of DAVIS pixels containing events, where a disparity estimate was also extracted.

The fan sequence is ideal for testing the ability of the algorithm to operate on rapidly moving objects, where standard frame-based algorithms would fail. Varying orientations of the revolving fan add continuously varying depth gradient to the dataset. Ground truth is extracted in terms of the plane in 3D space representing the blades' plane of rotation (Figure 6.5m). The butterfly sequence tests the ability of the algorithm to operate on irregular/unstructured objects which are rapidly rotating in a circular plane approximately perpendicular to the y-axis. Standard frame-based cameras have trouble extracting sharp features from this object. Ground truth is extracted in terms of the coordinates of the circle spanned by the rotating butterfly (Figure 6.5s).

6.4.2 Results

The RDS is tested on a model using 3×5 filters, left-right consistency constraints, no morphological erosion/dilation after rectification, and 31 disparity levels (0-30) plus a 'no-disparity' indicator disparity used to denote that no disparities were discovered (often due to self-occlusions). We also experiment with a post-processing phase with erosion and dilation applied to output disparity map in order to better regularize the output. Average disparity error and average recall before regularization is 0.1867/0.6572 and post-regularization is 0.0407/0.6305. We observe major improvements due to the regularization, often occurring in self-occluded regions. Errors increase in slanted regions due to foreshortening effects. The left-right consistency constraint is often sufficient to avoid false predictions in those regions.

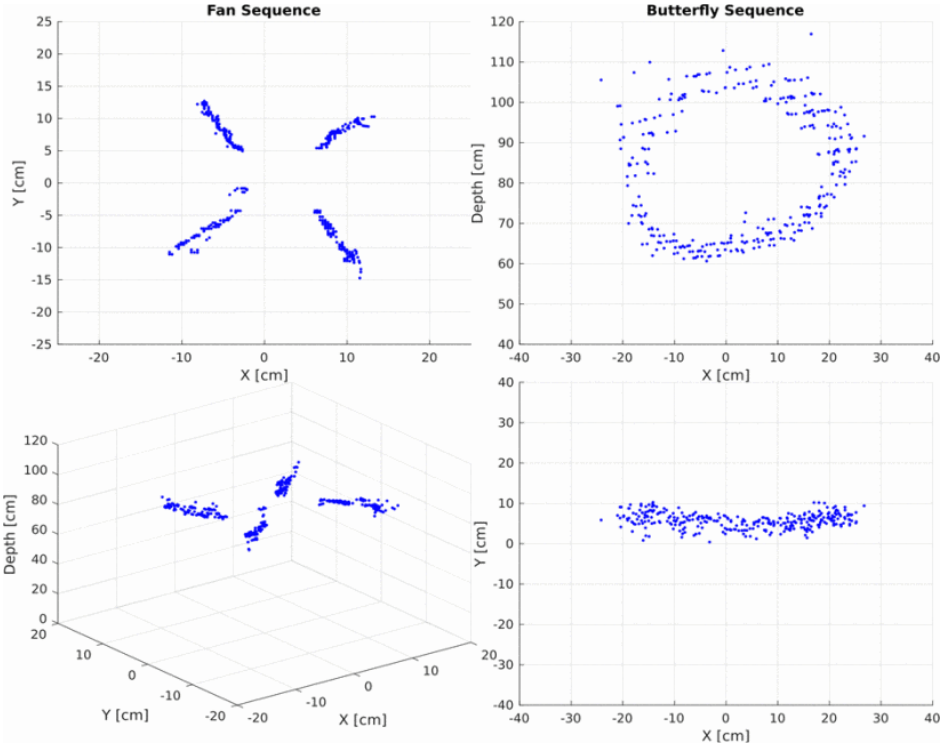


Figure 6.6: Depth reconstruction of the fan (first column) and butterfly sequence (second column), each shown from two viewpoints. Each point in the butterfly sequence shown is the median coordinate estimate of the butterfly location at a distinct time instant.

The evaluation on the non-synthetic dataset was done under the practical constraints of the availability of a limited number of NS1e boards on which non-simulated models could be

ran, as well as the need to process the full 240×180 DAVIS inputs at as high of a frame rate as possible. The models that run on live DAVIS input are operated at spike injection rate of up to 2000 Hz (a new input every 1/2000 seconds) and disparity map throughput of 400 Hz at a 0.5 ms tick period (400 distinct disparity maps produced every second) across a cluster of 9 TrueNorth chips. Running a model at the full 2000 Hz throughput comes at the expense of an increased neuron count. By adding a multiplexing spiking circuit inside the corelet, we are able to reuse each feature-extraction/WTA circuit to process the disparities for 5 different pixels, effectively decreasing the maximum disparity map throughput from 2000 Hz to 400 Hz, but at a lower neuron cost to process the full image (9 TrueNorth chips). However, we tested the maximum disparity map throughput achievable when running on TrueNorth chips, by executing a one-chip model on an input pixel subset, with no multiplexing (one disparity map ejected per tick) at a 0.5 ms tick period, achieving the 2000 Hz disparity map throughput. Reconstruction results on the fan and butterfly sequences are depicted in Figure 6.6.

6.5 Discussion

By using a spiking neural network, with low-precision weights, we have shown that the system is capable of injecting event streams and ejecting disparity maps at up to 2,000 Hz, at extremely low latencies (up to 22 ms on the systems tested) and low power. We have demonstrated the scalability of the system by using a scale-out approach that tiles multiple TrueNorth chips to process in parallel larger inputs. The system is highly parameterizable and can operate with other event based sensors such as ATIS (Posch et al., 2011) or DVS (Lichtsteiner et al., 2006).

Table 6.1 compares our approach with the existing literature in event based disparity. The main advantages of our approach regarding the state of the art are generalization to any

Table 6.1: Comparison of event-based depth estimation literature

Approaches	Ours	Osswald Osswald et al. (2017)	DikovDikov et al. (2017)	SchramlSchraml et al. (2007)	SchramlSchraml et al. (2015)	PiatkowskaPiatkowska et al. (2017)	EibensteinerEibensteiner et al. (2014)	MahowaldMahowald (1992)	RogisterRogister et al. (2012)	CamuñasCamuñas-Mesa et al. (2014)
Features of disparity algorithm and implementation										
Real time depth from live sensor input	X			X	X				X	X
Tested on long sequences from live sensor	X									
Real time w/ objects tilted in depth	X							X		X
Offline w/ objects tilted in depth	X	X						X		X
Real time w/ motion in depth	X		X	X						
Offline w/ motion in depth	X	X	X							
Scene independent throughput & latency	X			X	X		X		X	X
Fixed camera setup	X	X	X	X		X	X	X	X	X
No assumption of depth continuity	X			X	X			X	X	X
Fully neuromorphic disparity computation	X							X		
Neuromorphic rectification of input spikes	X									
Spike based algorithm	X	X	X		X	X	X	X	X	
Multi-resolution disparity computation	X							X		
Left to right consistency	X			X	X		X		X	
Uses event polarity compatibility	X	X					X	NA	X	X
Tested on dense RDS data	X	X						X		
Quant. eval. w/ objects moving in depth	X	X				X	X			
Tested on both fast and slow motions	X		X							
Implementation metrics										
Algorithm implementation hardware	Neuro	FPGA	Neuro	DSP	CPU	CPU	FPGA	ASIC	CPU	FPGA
Energy consumption (mWatts/Pixel)	0.021	-	16	0.30	-	-	-	-	-	-
Frames/s w/ objects tilted in depth	≤ 2000	1	-	-	-	-	-	40	-	20
Frames/s wo/ objects tilted in depth	≤ 2000	33	500	200	10	-	1140	>40	3333	20
Latency w/ objects tilted in depth (ms)	≤ 22	1000	-	-	-	-	-	25	-	50
Latency wo/ objects tilted in depth (ms)	≤ 22	30	2	-	-	-	0.87	<25	0.3	50
Max. image size tested (pixels)	43200	16384	11236	16384	1.4 M	-	16384	57	16384	16384
Max. disparity levels tested	41	21	32	-	-	-	36	9	128	-

depth map and object speed in real time, multi-resolution disparity calculation, scalability to live sensor feed with large input sizes and long sequences, and evaluation using synthetic as well as real world fast movements and depth gradients. The implemented neuromophic stereo disparity system achieves these advantages, while improving throughput up to four times and consuming 761 times less power per pixel regarding the most relevant state-of-the-art (Dikov et al., 2017). Furthermore, the homogeneous computational substrate stands in contrast to most of the other work in stereo for event-based inputs, by providing the first

example of a fully end-to-end low-power, high frame rate fully event-based neuromorphic stereo system capable of running on live input event streams.

Chapter 7

Conclusion

7.1 Summary

The primate visual system, including the retina, thalamus, visual cortex, and other brain areas, is a sophisticated device for sensing and perception of the environment. It results in fast and diverse visually guided behaviors. Understanding the computations performed by the deep hierarchies of the visual system and their information representation principles will benefit better algorithm development in terms of performance and computational efficiency (Kruger et al., 2012). Both of these qualities are highly desirable for algorithms to be deployed on autonomous systems and fast video monitoring in the real world.

This dissertation presents neural network models inspired by the primate visual system in order to understand the computations performed by the brain for motion perception and tracking and demonstrates the benefits of incorporating brain-like computations into vision algorithms. The ego-motion basis learning mechanism presented in Chapter 3 demonstrated how an overcomplete basis set can be derived for sparse representation of a high dimensional signal from noisy data, which is consistent with many aspects of neural coding in visual and

other sensory systems and has practical applications in signal analysis and recognition (Koniusz et al., 2013; Aharon et al., 2006). The neural models presented in Chapters 4 and 5 point toward the usefulness of goal driven training in modeling cortical neuron responses and population behavior. In Chapter 4, the MSTd-like neuron responses emerged in the deep layers of a neural network trained to accurately predict object and ego-motion. Similarly in Chapter 5, a recurrent neuron population with spontaneous firing activities like FEF neurons could generate the predictive pursuit behaviors observed in humans. Chapter 6 demonstrates incorporating brain inspired spike based data representation enables a high throughput stereo vision system for dynamic scenes using a tiny fraction of energy used by traditional computers.

Overall, these models and implementations combine knowledge from neuroscience and computer vision in an interdisciplinary study of vision. The present work might be of interest to both research communities as it provides i) an intrinsic optic flow decomposition into description of dynamic scene elements analogous to higher level motion processing in the brain, ii) a neural representation scheme of motion components by the dorsal visual pathway that can be verified experimentally, iii) a mechanistic account of adaptive internal model based predictive smooth pursuit eye movements that is useful for long term object tracking with occlusions, and iv) a demonstration of implementation level description of vision system using neuromorphic hardware. We hope future studies along this direction will lead to algorithms as efficient and accurate as the visual cortex to be embodied into intelligent mobile robots.

7.2 Future Directions

The neural network models presented in this dissertation could be combined together to create a unified model of visual motion perception and tracking in cluttered dynamic envi-

ronments. The MSTd neurons in the cortex project to the FEF area that controls predictive pursuit eye movements to keep the moving target on high acuity fovea (Ilg and Schumann, 2007). Therefore, a unified model will provide a complete account of the computations along the pursuit pathway.

Another future research could be to analyze the individual neuron responses in the pursuit model in relation to FEF neuron responses during smooth pursuit eye movement to understand how spontaneous neuron activities generate the predictive pursuit behaviors observed in humans and other primates. The pursuit model could also be implemented on a robotic platform to demonstrate human-like visual object tracking in the real world. For example, self-driving vehicles can use this model to predict the future trajectories of other vehicles and pedestrians for planning their own maneuvers.

The CNN model of motion perception in the dorsal visual pathway could be tested on more naturalistic stimuli. This could show differences in neuron responses to those observed for synthetic stimuli with lesser variations (Sato et al., 2010) and could predict how the MSTd neurons might behave in response to real world scene depth, object, and ego-motion. This could also guide future neurophysiology experiment design to investigate cortical motion perception under naturalistic conditions. The object and ego-motion estimation methods developed here using sparse representations could be applied to compression and other video processing applications (Furht et al., 2012; Li et al., 2018; Quelhas et al., 2005). Finally, the spiking stereo algorithm and implementation in Chapter 6 should inspire spiking implementations of the motion perception and tracking models on neuromorphic hardware for low power robotic applications.

Bibliography

Adelson, E. H. and Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Josa a*, 2(2):284–299.

Adelson, E. H. and Movshon, J. A. (1982). Phenomenal coherence of moving visual patterns. *Nature*, 300(5892):523–525.

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322.

Amir, A., Datta, P., Risk, W. P., Cassidy, A. S., Kusnitz, J. A., Esser, S. K., Andreopoulos, A., Wong, T. M., Flickner, M., Alvarez-Icaza, R., et al. (2013). Cognitive computing programming paradigm: a corelet language for composing networks of neurosynaptic cores. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–10. IEEE.

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. (2017). A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252.

Bahill, A. T. and McDonald, J. D. (1983). Model emulates human smooth pursuit system producing zero-latency target tracking. *Biological cybernetics*, 48(3):213–222.

Bak, A., Bouchafa, S., and Aubert, D. (2014). Dynamic objects detection through visual odometry and stereo-vision: a study of inaccuracy and improvement sources. *Machine Vision and Applications*, 25(3):681–697.

Barlow, H. B. (1981). The ferrier lecture, 1980: Critical limiting factors in the design of the eye and visual cortex. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pages 1–34.

Barnes, G. and Asselman, P. (1991). The mechanism of prediction in human smooth pursuit eye movements. *The Journal of physiology*, 439(1):439–461.

Barnes, G., Donnelly, S., and Eason, R. (1987). Predictive velocity estimation in the pursuit reflex response to pseudo-random and step displacement stimuli in man. *The Journal of Physiology*, 389(1):111–136.

Barnes, G. and Ruddock, C. (1989). Factors affecting the predictability of pseudo-random motion stimuli in the pursuit reflex of man. *The Journal of physiology*, 408(1):137–165.

Barnes, G. and Wells, S. (1999). Modelling prediction in ocular pursuit. In *Current oculomotor research*, pages 97–107. Springer.

Barnes, G. R. (2008). Cognitive processes involved in smooth pursuit eye movements. *Brain and cognition*, 68(3):309–326.

Barnett, V., Lewis, T., and Abeles, F. (1979). Outliers in Statistical Data. *Physics Today*, 32:73.

Beard, R. W., Kingston, D. B., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T. W., and Goodrich, M. A. (2005). Autonomous vehicle technologies for small fixed-wing uavs. *JACIC*, 2(1):92–108.

Belbachir, A. N., Schraml, S., Mayerhofer, M., and Hofstätter, M. (2014). A novel hdr depth camera for real-time 3d 360° panoramic vision. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 425–432. IEEE.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., Alvarez-Icaza, R., Arthur, J. V., Merolla, P. A., and Boahen, K. (2014). Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716.

Bennett, S. J. and Barnes, G. R. (2006). Smooth ocular pursuit during the transient disappearance of an accelerating visual target: the role of reflexive and voluntary control. *Experimental brain research*, 175(1):1–10.

Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852.

Beyeler, M., Dutt, N., and Krichmar, J. L. (2016). 3d visual response properties of mstd emerge from an efficient, sparse population code. *Journal of Neuroscience*, 36(32):8399–8415.

Black, M. J. and Anandan, P. (1996). The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

Braddick, O. J., O'Brien, J. M., Wattam-Bell, J., Atkinson, J., Hartley, T., and Turner, R. (2001). Brain areas sensitive to coherent visual motion. *Perception*, 30(1):61–72.

Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T. (2014). A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341.

Britten, K. H. (2008). Mechanisms of self-motion perception. *Annu. Rev. Neurosci.*, 31:389–410.

Brosch, T. and Neumann, H. (2016). Event-based optical flow on neuromorphic hardware. In *proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) on 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 551–558. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Browning, N. A., Grossberg, S., and Mingolla, E. (2009). A neural model of how the brain computes heading from optic flow in realistic scenes. *Cognitive psychology*, 59(4):320–356.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 611–625.

Byravan, A. and Fox, D. (2017). Se3-nets: Learning rigid body motion using deep neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 173–180.

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., and DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS Comput Biol*, 10(12):e1003963.

Campbell, J., Sukthankar, R., Nourbakhsh, I., and Pahwa, A. (2005). A robust visual odometry and precipice detection system using consumer-grade monocular vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3421–3427.

Camuñas-Mesa, L. A., Serrano-Gotarredona, T., Ieng, S. H., Benosman, R. B., and Linares-Barranco, B. (2014). On the use of orientation filters for 3d reconstruction in event-driven stereo vision. *Frontiers in neuroscience*, 8.

Cassidy, A. S., Merolla, P., Arthur, J. V., Esser, S. K., Jackson, B., Alvarez-Icaza, R., Datta, P., Sawada, J., Wong, T. M., Feldman, V., et al. (2013). Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–10. IEEE.

Chou, I.-h. and Lisberger, S. G. (2004). The role of the frontal pursuit area in learning in smooth pursuit eye movements. *The Journal of neuroscience*, 24(17):4124–4133.

Choudrey, R. A. (2002). *Variational methods for Bayesian independent component analysis*. PhD thesis, University of Oxford.

Cichy, R. M., Khosla, A., Pantazis, D., and Oliva, A. (2017). Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. *NeuroImage*, 153:346–358.

Collewijn, H. and Tamminga, E. P. (1984). Human smooth and saccadic eye movements during voluntary pursuit of different target motions on different backgrounds. *The Journal of Physiology*, 351(1):217–250.

Concha, A. and Civera, J. (2015). DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5686–5693.

Dallos, P. and Jones, R. (1963). Learning behavior of the eye fixation control system. *IEEE Transactions on Automatic Control*, 8(3):218–227.

Dawson, M. R. (1998). *Understanding cognitive science*. Blackwell Oxford.

de Brouwer, S., Missal, M., Barnes, G., and Lefèvre, P. (2002). Quantitative analysis of catch-up saccades during sustained pursuit. *Journal of neurophysiology*, 87(4):1772–1780.

de Xivry, J.-J. O., Coppe, S., Blohm, G., and Lefevre, P. (2013). Kalman filtering naturally accounts for visually guided and predictive smooth pursuit dynamics. *Journal of Neuroscience*, 33(44):17301–17313.

DeAngelis, G. C. and Newsome, W. T. (1999). Organization of disparity-selective neurons in macaque area mt. *Journal of Neuroscience*, 19(4):1398–1415.

Deno, D., Crandall, W., Sherman, K., and Keller, E. (1995). Characterization of prediction in the primate visual smooth pursuit system. *Biosystems*, 34(1-3):107–128.

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415–434.

Dicke, P. W., Barash, S., Ilg, U. J., and Thier, P. (2004). Single-neuron evidence for a contribution of the dorsal pontine nuclei to both types of target-directed eye movements, saccades and smooth-pursuit. *European Journal of Neuroscience*, 19(3):609–624.

Dikov, G., Firouzi, M., Röhrbein, F., Conradt, J., and Richter, C. (2017). Spiking cooperative stereo-matching at 2 ms latency with neuromorphic hardware. In *Conference on Biomimetic and Biohybrid Systems*, pages 119–137. Springer.

Dobkins, K. R. and Teller, D. Y. (1996). Infant contrast detectors are selective for direction of motion. *Vision Research*, 36(2):281–294.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2005). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18.

Duffy, C. J. and Wurtz, R. H. (1991a). Sensitivity of MST neurons to optic flow stimuli. I. A continuum of response selectivity to large-field stimuli. *Journal of neurophysiology*, 65(6):1329–1345.

Duffy, C. J. and Wurtz, R. H. (1991b). Sensitivity of mst neurons to optic flow stimuli. ii. mechanisms of response selectivity revealed by small-field stimuli. *Journal of neurophysiology*, 65(6):1346–1359.

Dursteler, M. and Wurtz, R. H. (1988). Pursuit and optokinetic deficits following chemical lesions of cortical areas mt and mst. *Journal of Neurophysiology*, 60(3):940–965.

Eckmiller, R. and Mackeben, M. (1978). Pursuit eye movements and their neural control in the monkey. *Pflügers Archiv European Journal of Physiology*, 377(1):15–23.

Eibensteiner, F., Kogler, J., and Scharinger, J. (2014). A high-performance hardware architecture for a frameless stereo vision algorithm implemented on a fpga platform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 623–630.

Eifuku, S. and Wurtz, R. H. (1998). Response to motion in extrastriate area mstl: center-surround interactions. *Journal of Neurophysiology*, 80(1):282–296.

Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745.

Engel, J., Schöps, T., and Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision*, pages 834–849.

Essen, D. v. and Zeki, S. (1978). The topographic organization of rhesus monkey prestriate cortex. *The Journal of physiology*, 277(1):193–226.

Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, A., Berg, D. J., McKinstry, J. L., Melano, T., Barch, D. R., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences*, page 201604850.

Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47.

Fredriksson, J., Larsson, V., and Olsson, C. (2015). Practical robust two-view translation estimation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2684–2690.

Froehler, M. T. and Duffy, C. J. (2002). Cortical neurons encoding path and place: where you go is where you are. *Science*, 295(5564):2462–2465.

Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130.

Fukushima, K., Yamanobe, T., Shinmei, Y., and Fukushima, J. (2002). Predictive responses of periarcuate pursuit neurons to visual target motion. *Experimental brain research*, 145(1):104–120.

Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., and Brown, A. D. (2013). Overview of the spinnaker system architecture. *IEEE Transactions on Computers*, 62(12):2454–2467.

Furht, B., Greenberg, J., and Westwater, R. (2012). *Motion estimation algorithms for video compression*, volume 379. Springer Science & Business Media.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Towards internet-scale multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1434–1441.

Gaidon, A., Wang, Q., Cabon, Y., and Vig, E. (2016). Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4340–4349.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

Gaymard, B., Pierrot-Deseilligny, C., Rivaud, S., and Velut, S. (1993). Smooth pursuit eye movement deficits after pontine nuclei lesions in humans. *Journal of Neurology, Neurosurgery & Psychiatry*, 56(7):799–807.

Geesaman, B. J. and Andersen, R. A. (1996). The analysis of complex motion patterns by form/cue invariant mstd neurons. *Journal of Neuroscience*, 16(15):4716–4732.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361.

Geman, S. (1999). Hierarchy in machine and natural vision. In *Proceedings of the Scandinavian Conference on Image Analysis*, volume 1, pages 179–184.

Giachetti, A., Campani, M., and Torre, V. (1998). The use of optical flow for road navigation. *IEEE Transactions on Robotics and Automation*, 14(1):34–48.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Glickstein, M., Gerrits, N., Kralj-Hans, I., Mercier, B., Stein, J., and Voogd, J. (1994). Visual pontocerebellar projections in the macaque. *Journal of Comparative Neurology*, 349(1):51–72.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 315–323.

Godard, C., Mac Aodha, O., Firman, M., and Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3838.

Graziano, M. S., Andersen, R. A., and Snowden, R. J. (1994). Tuning of mst neurons to spiral motions. *Journal of Neuroscience*, 14(1):54–67.

Grill-Spector, K. and Weiner, K. S. (2014). The functional architecture of the ventral temporal cortex and its role in categorization. *Nature Reviews Neuroscience*, 15(8):536–548.

Gross, C. G., Rodman, H. R., Gochin, P. M., and Colombo, M. W. (1993). Inferior temporal cortex as a pattern recognition device. In *Computational Learning & Cognition: Proceedings of the Third NEC Research Symposium*, pages 44–73. Soc for Industrial & Applied Math.

Grossberg, S., Léveillé, J., and Versace, M. (2011). How do object reference frames and motion vector decomposition emerge in laminar cortical circuits? *Attention, Perception, & Psychophysics*, 73(4):1147–1170.

Grossberg, S., Mingolla, E., and Pack, C. (1999). A neural model of motion processing and visual navigation by cortical area mst. *Cerebral Cortex*, 9(8):878–895.

Güçlü, U. and van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014.

Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hedges, J. H., Gartshteyn, Y., Kohn, A., Rust, N. C., Shadlen, M. N., Newsome, W. T., and Movshon, J. A. (2011). Dissociation of neuronal and psychophysical responses to local and global motion. *Current Biology*, 21(23):2023–2028.

Heeger, D. J. and Jepson, A. D. (1992). Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Houenou, A., Bonnifait, P., Cherfaoui, V., and Yao, W. (2013). Vehicle trajectory prediction based on motion model and maneuver recognition. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, pages 4363–4369.

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5(Nov):1457–1469.

Hu, J., Ma, H., Zhu, S., Li, P., Xu, H., Fang, Y., Chen, M., Han, C., Fang, C., Cai, X., et al. (2018). Visual motion processing in macaque v2. *Cell reports*, 25(1):157–167.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.

Huber, P. J. (2004). *Robust Statistics*. John Wiley & Sons.

Hughes, A. E., Greenwood, J. A., Finlayson, N. J., and Schwarzkopf, D. S. (2019). Population receptive field estimates for motion-defined stimuli. *NeuroImage*, 199:245–260.

Ilg, U. J. and Schumann, S. (2007). Primate area mst-l is involved in the generation of goal-directed eye and hand movements. *Journal of Neurophysiology*, 97(1):761–771.

Indiveri, G., Chicca, E., and Douglas, R. (2006). A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE transactions on neural networks*, 17(1):211–221.

Inman, H. F. and Bradley Jr, E. L. (1989). The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics-Theory and Methods*, 18(10):3851–3874.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13.

Jaegle, A., Phillips, S., and Daniilidis, K. (2016). Fast, robust, continuous monocular egomotion computation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 773–780.

Jaimez, M., Kerl, C., Gonzalez-Jimenez, J., and Cremers, D. (2017). Fast odometry and scene flow from RGB-D cameras based on geometric clustering. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3992–3999.

Jiang, N., Rong, W., Peng, B., Nie, Y., and Xiong, Z. (2015). An empirical analysis of different sparse penalties for autoencoder in unsupervised feature learning. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer.

Kak, S. (2016). Generalized unary coding. *Circuits, Systems, and Signal Processing*, 35(4):1419–1426.

Keating, E. (1991). Frontal eye field lesions impair predictive and visually-guided pursuit eye movements. *Experimental Brain Research*, 86(2):311–323.

Keating, E. G. (1993). Lesions of the frontal eye field impair pursuit eye movements, but preserve the predictions driving them. *Behavioural brain research*, 53(1):91–104.

Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11).

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kogler, J., Sulzbachner, C., Humenberger, M., and Eibensteiner, F. (2011). Address-event based stereo vision with bio-inspired silicon retina imagers. In *Advances in theory and applications of stereo vision*. InTech.

Komatsu, H. and Wurtz, R. H. (1988). Relation of cortical areas mt and mst to pursuit eye movements. i. localization and visual properties of neurons. *Journal of neurophysiology*, 60(2):580–603.

Koniusz, P., Yan, F., and Mikolajczyk, K. (2013). Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *Computer vision and image understanding*, 117(5):479–492.

Krauzlis, R. J. and Lisberger, S. G. (1994). A model of visually-guided smooth pursuit eye movements based on behavioral observations. *Journal of computational neuroscience*, 1(4):265–283.

Krekelberg, B. and Albright, T. D. (2005). Motion mechanisms in macaque mt. *Journal of neurophysiology*, 93(5):2908–2921.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., Rodriguez-Sanchez, A. J., and Wiskott, L. (2012). Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1847–1871.

Layton, O. W. and Fajen, B. R. (2020). Computational mechanisms for perceptual stability using disparity and motion parallax. *Journal of Neuroscience*, 40(5):996–1014.

Layton, O. W. and Niehorster, D. (2019). A model of how depth facilitates scene-relative object motion perception. *PLoS computational biology*, 15(11).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.

Lee, M. and Fowlkes, C. C. (2019). CeMNet: Self-supervised learning for accurate continuous ego-motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Lencer, R., Nagel, M., Sprenger, A., Zapf, S., Erdmann, C., Heide, W., and Binkofski, F. (2004). Cortical mechanisms of smooth pursuit eye movements with target blanking. an fmri study. *European Journal of Neuroscience*, 19(5):1430–1436.

Levinson, E. and Sekuler, R. (1975). The independence of channels in human vision selective for direction of movement. *The Journal of Physiology*, 250(2):347–366.

Lewicki, M. S. and Sejnowski, T. J. (2000). Learning overcomplete representations. *Neural Computation*, 12(2):337–365.

Li, M., Xie, Q., Zhao, Q., Wei, W., Gu, S., Tao, J., and Meng, D. (2018). Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653.

Lichtsteiner, P., Posch, C., and Delbruck, T. (2006). A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pages 2060–2069. IEEE.

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128×128 120 db 15μ s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576.

Lindsay, G. (2020). Convolutional neural networks as a model of the visual system: past, present, and future. *Journal of Cognitive Neuroscience*, pages 1–15.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Longuet-Higgins, H. C. and Prazdny, K. (1980). The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397.

Louizos, C., Welling, M., and Kingma, D. P. (2017). Learning sparse neural networks through L_0 regularization. *arXiv preprint arXiv:1712.01312*.

Lu, H. D., Chen, G., Tanigawa, H., and Roe, A. W. (2010). A motion direction map in macaque v2. *Neuron*, 68(5):1002–1013.

Lv, Z., Kim, K., Troccoli, A., Sun, D., Rehg, J. M., and Kautz, J. (2018). Learning rigidity in dynamic scenes with a moving camera for 3D motion field estimation. In *Proceedings of the European Conference on Computer Vision*, pages 468–484.

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560.

MacAvoy, M. G., Gottlieb, J. P., and Bruce, C. J. (1991). Smooth-pursuit eye movement representation in the primate frontal eye field. *Cerebral Cortex*, 1(1):95–102.

Mahjourian, R., Wicke, M., and Angelova, A. (2018). Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675.

Mahowald, M. (1992). *VLSI analogs of neuronal visual processing: a synthesis of form and function*. PhD thesis, California Institute of Technology.

Marr, D. and Poggio, T. (1976). From understanding computation to understanding neural circuitry. Technical report, Massachusetts Institute of Technology.

Marr, D., Poggio, T., et al. (1976). Cooperative computation of stereo disparity. *From the Retina to the Neocortex*, pages 239–243.

Matsumiya, K. and Ando, H. (2009). World-centered perception of 3d object motion during visually guided self-motion. *Journal of Vision*, 9(1):15–15.

Maunsell, J. H. and Van Essen, D. C. (1983). Functional properties of neurons in middle temporal visual area of the macaque monkey. i. selectivity for stimulus direction, speed, and orientation. *Journal of neurophysiology*, 49(5):1127–1147.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048.

Medina, J. F. and Lisberger, S. G. (2009). Encoding and decoding of learned smooth-pursuit eye movements in the floccular complex of the monkey cerebellum. *Journal of neurophysiology*, 102(4):2039–2054.

Menze, M. and Geiger, A. (2015). Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673.

Meunier, D., Lambiotte, R., and Bullmore, E. T. (2010). Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience*, 4:200.

Miller, A. (2002). *Subset Selection in Regression*. Chapman and Hall/CRC.

Mishkin, M. and Ungerleider, L. G. (1982). Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural brain research*, 6(1):57–77.

Mishkin, M., Ungerleider, L. G., and Macko, K. A. (1983). Object vision and spatial vision: two cortical pathways. *Trends in neurosciences*, 6:414–417.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Neumann, J. v. (1958). *The computer and the brain*. Yale University Press.

Newcombe, F., Ratcliff, G., and Damasio, H. (1987). Dissociable visual and spatial impairments following right posterior cerebral lesions: Clinical, neuropsychological and anatomical evidence. *Neuropsychologia*, 25(1):149–161.

Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In *Proceedings of the International Conference on Computer Vision*, pages 2320–2327.

Nishida, S., Kawabe, T., Sawayama, M., and Fukiage, T. (2018). Motion perception: From detection to interpretation. *Annual review of vision science*, 4:501–523.

Nishimoto, S. and Gallant, J. L. (2011). A three-dimensional spatiotemporal receptive field model explains responses of area mt neurons to naturalistic movies. *Journal of Neuroscience*, 31(41):14551–14564.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):0756–777.

Nyffeler, T., Rivaud-Pechoux, S., Wattiez, N., and Gaymard, B. (2008). Involvement of the supplementary eye field in oculomotor predictive behavior. *Journal of Cognitive Neuroscience*, 20(9):1583–1594.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325.

Orban, G., Kennedy, H., and Bullier, J. (1986). Velocity sensitivity and direction selectivity of neurons in areas v1 and v2 of the monkey: influence of eccentricity. *Journal of Neurophysiology*, 56(2):462–480.

Orban, G. A. (2008). Higher order visual processing in macaque extrastriate cortex. *Physiological reviews*, 88(1):59–89.

Osswald, M., Ieng, S.-H., Benosman, R., and Indiveri, G. (2017). A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems. *Scientific reports*, 7:40703.

Park, K., Jabri, M., Lee, S.-Y., and Sejnowski, T. J. (2000). Independent components of optical flows have mstd-like receptive fields. In *International Workshop on Independent Component Analysis and Blind Signal Separation*.

Piatkowska, E., Kogler, J., Belbachir, N., and Gelautz, M. (2017). Improved cooperative stereo matching for dynamic vision sensors with ground truth evaluation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 370–377. IEEE.

Posch, C., Matolin, D., and Wohlgenannt, R. (2011). A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275.

Priebe, N. J., Lisberger, S. G., and Movshon, J. A. (2006). Tuning for spatiotemporal frequency and speed in directionally selective neurons of macaque striate cortex. *Journal of Neuroscience*, 26(11):2941–2950.

Quelhas, P., Monay, F., Odobez, J.-M., Gatica-Perez, D., Tuytelaars, T., and Van Gool, L. (2005). Modeling scenes with local descriptors and latent aspects. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 883–890. IEEE.

Quiroga, J., Brox, T., Devernay, F., and Crowley, J. (2014). Dense semi-rigid scene flow estimation from RGBD images. In *Proceedings of the European Conference on Computer Vision*, pages 567–582.

Rambold, H., Churchland, A., Selig, Y., Jasmin, L., and Lisberger, S. (2002). Partial ablations of the flocculus and ventral paraflocculus in monkeys cause linked deficits in smooth pursuit eye movements and adaptive modification of the vor. *Journal of Neurophysiology*, 87(2):912–924.

Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., and Black, M. J. (2019). Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12240–12249.

Rasche, C. and Gegenfurtner, K. R. (2009). Precision of speed discrimination and smooth pursuit eye movements. *Vision research*, 49(5):514–523.

Raymond, J. and Braddick, O. (1996). Responses to opposed directions of motion:: Continuum or independent mechanisms? *Vision research*, 36(13):1931–1937.

Recanzone, G., Wurtz, R., and Schwarz, U. (1997). Responses of mt and mst neurons to one and two moving objects in the receptive field. *Journal of neurophysiology*, 78(6):2904–2915.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Reichardt, W. (1957). Autokorrelations-auswertung als funktionsprinzip des zentralnervensystems. *Zeitschrift für Naturforschung B*, 12(7):448–457.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025.

Riesenhuber, M. and Poggio, T. (2000). Computational models of object recognition in cortex: A review. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB.

Robinson, D. A., Gordon, J., and Gordon, S. (1986). A model of the smooth pursuit eye movement system. *Biological cybernetics*, 55(1):43–57.

Rogister, P., Benosman, R., Ieng, S.-H., Lichtsteiner, P., and Delbruck, T. (2012). Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):347–353.

Roy, J. E. and Cullen, K. E. (2003). Brain stem pursuit pathways: dissociating visual, vestibular, and proprioceptive inputs during combined eye-head gaze tracking. *Journal of neurophysiology*, 90(1):271–290.

Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation: The basic theory. *Backpropagation: Theory, architectures and applications*, pages 1–34.

Rust, N. C., Mante, V., Simoncelli, E. P., and Movshon, J. A. (2006). How mt cells analyze the motion of visual patterns. *Nature neuroscience*, 9(11):1421–1431.

Sasaki, R., Angelaki, D. E., and DeAngelis, G. C. (2017). Dissociation of self-motion and object motion by linear population decoding that approximates marginalization. *Journal of Neuroscience*, 37(46):11204–11219.

Sasaki, R., Angelaki, D. E., and DeAngelis, G. C. (2019). Processing of object motion and self-motion in the lateral subdivision of the medial superior temporal area in macaques. *Journal of neurophysiology*, 121(4):1207–1221.

Sato, N., Kishore, S., Page, W. K., and Duffy, C. J. (2010). Cortical neurons combine visual cues about self-movement. *Experimental brain research*, 206(3):283–297.

Sawada, J., Akopyan, F., Cassidy, A. S., Taba, B., Debole, M. V., Datta, P., Alvarez-Icaza, R., Amir, A., Arthur, J. V., Andreopoulos, A., et al. (2016). Truenorth ecosystem for brain-inspired computing: scalable systems, software, and applications. In *High Performance Computing, Networking, Storage and Analysis, SC16: International Conference for*, pages 130–141. IEEE.

Schemmel, J., Briiderle, D., Griibl, A., Hock, M., Meier, K., and Millner, S. (2010). A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Circuits and systems (ISCAS), proceedings of 2010 IEEE international symposium on*, pages 1947–1950. IEEE.

Schraml, S., Belbachir, A. N., and Bischof, H. (2016). An event-driven stereo system for real-time 3-d 360° panoramic vision. *IEEE Transactions on Industrial Electronics*, 63(1):418–428.

Schraml, S., Belbachir, A. N., Milosevic, N., and Schön, P. (2010). Dynamic stereo vision system for real-time tracking. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1409–1412. IEEE.

Schraml, S., Nabil Belbachir, A., and Bischof, H. (2015). Event-driven stereo matching for real-time 3d panoramic vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 466–474.

Schraml, S., Schön, P., and Milosevic, N. (2007). Smartcam for real-time stereo vision-address-event based embedded system. In *VISAPP (2)*, pages 466–471.

Seeliger, K., Fritsche, M., Güçlü, U., Schoenmakers, S., Schoffelen, J.-M., Bosch, S., and Van Gerven, M. (2018). Convolutional neural network-based encoding and decoding of visual object recognition in space and time. *NeuroImage*, 180:253–266.

Serre, T. (2014). Hierarchical models of the visual system.

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., and Poggio, T. (2007). A quantitative theory of immediate visual recognition. *Progress in brain research*, 165:33–56.

Shibata, T., Tabata, H., Schaal, S., and Kawato, M. (2005). A model of smooth pursuit in primates based on learning the target dynamics. *Neural Networks*, 18(3):213–224.

Shichinohe, N., Akao, T., Kurkin, S., Fukushima, J., Kaneko, C. R., and Fukushima, K. (2009). Memory and decision making in the frontal cortex during visual motion processing for smooth pursuit eye movements. *Neuron*, 62(5):717–732.

Shidara, M., Kawano, K., Gomi, H., and Kawato, M. (1993). Inverse-dynamics model eye movement control by purkinje cells in the cerebellum. *Nature*, 365(6441):50–52.

Simoncelli, E., Freeman, W., Adelson, E., and Heeger, D. (1992). Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607.

Simoncelli, E. P. and Heeger, D. J. (1998). A model of neuronal responses in visual area mt. *Vision research*, 38(5):743–761.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Spering, M. and Gegenfurtner, K. R. (2007). Contextual effects on smooth-pursuit eye movements. *Journal of Neurophysiology*, 97(2):1353–1367.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stanton, G., Goldberg, M., and Bruce, C. (1988). Frontal eye field efferents in the macaque monkey: Ii. topography of terminal fields in midbrain and pons. *Journal of Comparative Neurology*, 271(4):493–506.

Stein, G. P., Mano, O., and Shashua, A. (2000). A robust method for computing vehicle ego-motion. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 362–368.

Strasdat, H., Montiel, J., and Davison, A. J. (2010). Scale drift-aware large scale monocular SLAM. *Robotics: Science and Systems VI*, 2(3):7.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018). PWC-net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943.

Sunkara, A., DeAngelis, G. C., and Angelaki, D. E. (2016). Joint representation of translational and rotational components of optic flow in parietal cortex. *Proceedings of the National Academy of Sciences*, 113(18):5077–5082.

Sussillo, D. and Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557.

Sussillo, D., Toyoizumi, T., and Maass, W. (2007). Self-tuning of neural circuits through short-term synaptic plasticity. *Journal of neurophysiology*, 97(6):4079–4095.

Takagi, M., Zee, D. S., and Tamargo, R. J. (2000). Effects of lesions of the oculomotor cerebellar vermis on eye movements in primate: smooth pursuit. *Journal of Neurophysiology*, 83(4):2047–2062.

Takahashi, K., Gu, Y., May, P. J., Newlands, S. D., DeAngelis, G. C., and Angelaki, D. E. (2007). Multimodal coding of three-dimensional rotation and translation in area MSTd: comparison of visual and vestibular selectivity. *Journal of Neuroscience*, 27(36):9742–9756.

Tanaka, K. and Saito, H.-A. (1989). Analysis of motion of the visual field by direction, expansion/contraction, and rotation cells clustered in the dorsal part of the medial superior temporal area of the macaque monkey. *Journal of neurophysiology*, 62(3):626–641.

Teney, D. and Hebert, M. (2016). Learning to extract motion from videos in convolutional neural networks. In *Asian Conference on Computer Vision*, pages 412–428. Springer.

Thiele, A., Dobkins, K. R., and Albright, T. D. (2000). Neural correlates of contrast detection at threshold. *Neuron*, 26(3):715–724.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tripp, B. P. (2017). Similarities and differences between stimulus tuning in the inferotemporal visual cortex and convolutional networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3551–3560. IEEE.

Tseng, P. (2009). Further results on stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 55(2):888–899.

Tung, H.-Y. F., Harley, A. W., Seto, W., and Fragkiadaki, K. (2017). Adversarial inverse graphics networks: Learning 2D-to-3D lifting and image-to-image translation from unpaired supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4364–4372.

Ungerleider, L. G. and Desimone, R. (1986). Cortical connections of visual area mt in the macaque. *Journal of Comparative Neurology*, 248(2):190–222.

Van den Berg, A. (1988). Human smooth pursuit during transient perturbations of predictable and unpredictable target movement. *Experimental Brain Research*, 72(1):95.

Van Essen, D. C., Drury, H. A., Dickson, J., Harwell, J., Hanlon, D., and Anderson, C. H. (2001). An integrated software suite for surface-based analyses of cerebral cortex. *Journal of the American Medical Informatics Association*, 8(5):443–459.

Van Santen, J. P. and Sperling, G. (1985). Elaborated reichardt detectors. *JOSA A*, 2(2):300–321.

Varjú, D. and Schnitzler, H.-U. (2012). *Localization and orientation in biology and engineering*. Springer Science & Business Media.

Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. (1999). Three-dimensional scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 722–729.

Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., and Fragkiadaki, K. (2017). SfM-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*.

Wang, J. Y. and Adelson, E. H. (1994). Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638.

Warren, P. A. and Rushton, S. K. (2007). Perception of object trajectory: parsing retinal motion into self and object movement components. *Journal of Vision*, 7(11):2–2.

Warren, P. A. and Rushton, S. K. (2009). Optic flow processing for the assessment of object movement during ego movement. *Current Biology*, 19(18):1555–1560.

Watson, A. B., Thompson, P. G., Murphy, B. J., and Nachmias, J. (1980). Summation and discrimination of gratings moving in opposite directions. *Vision Research*, 20(4):341–347.

Whittaker, S. G. and Eaholtz, G. (1982). Learning patterns of eye motion for foveal pursuit. *Investigative ophthalmology & visual science*, 23(3):393–397.

Wohlberg, B. (2003). Noise sensitivity of sparse signal representations: Reconstruction error bounds for the inverse problem. *IEEE Transactions on Signal Processing*, 51(12):3053–3060.

Wulff, J., Sevilla-Lara, L., and Black, M. J. (2017). Optical flow in mostly rigid scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4671–4680.

Xiao, D.-K., Marcar, V., Raiguel, S., and Orban, G. (1997). Selectivity of macaque mt/v5 neurons for surface orientation in depth specified by motion. *European Journal of Neuroscience*, 9(5):956–964.

Xiao, Q., Barborica, A., and Ferrera, V. P. (2007). Modulation of visual responses in macaque frontal eye field during covert tracking of invisible targets. *Cerebral Cortex*, 17(4):918–928.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624.

Yang, Y., Liu, S., Chowdhury, S. A., DeAngelis, G. C., and Angelaki, D. E. (2011). Binocular disparity tuning and visual–vestibular congruency of multisensory neurons in macaque parietal cortex. *Journal of Neuroscience*, 31(49):17905–17916.

Yang, Z., Wang, P., Wang, Y., Xu, W., and Nevatia, R. (2018). Every pixel counts: Unsupervised geometry learning with holistic 3D motion understanding. *arXiv preprint arXiv:1806.10556*.

Yasui, S. and Young, L. (1984). On the predictive control of foveal eye tracking and slow phases of optokinetic and vestibular nystagmus. *The Journal of physiology*, 347(1):17–33.

Yin, Z. and Shi, J. (2018). GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zemel, R. S. and Sejnowski, T. J. (1995). Grouping components of three-dimensional moving objects in area mst of visual cortex. In *Advances in neural information processing systems*, pages 165–172.

Zhang, T. and Tomasi, C. (1999). Fast, robust, and consistent camera motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 164–170.

Zhang, T. and Tomasi, C. (2002). On the consistency of instantaneous rigid motion estimation. *International Journal of Computer Vision*, 46(1):51–79.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10.

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.